



刘朝华 著

混合免疫智能算法 理论及应用



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

国家自然科学基金项目
中国博士后科学基金面上项目
高校博士点基金项目
湖南省自然科学基金项目

混合免疫智能算法理论及应用

刘朝华 著

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书在全面总结国内外的混合免疫智能算法及其应用成果的基础上,着重介绍作者在混合免疫智能算法理论及应用这一领域的研究成果,主要包括竞争合作性协同免疫进化算法模型设计、算法分析与实验;免疫双态粒子群计算模型构建、算法分析与实验;免疫协同粒子群算法模型构建、算法分析与实验;免疫蚁群自适应融合算法设计及实验分析;混合免疫智能算法在复杂工业控制系统中的应用研究,如基于混合免疫智能算法的混沌系统自抗扰优化控制和基于混合免疫智能算法的永磁同步电机多参数辨识与状态监测等实际工业应用。本书为读者提供了混合免疫智能集成计算模型构建、算法设计、理论分析、算法编程和工程实际应用等新方法、新技术、新思路。

本书可供高等院校自动化、计算机科学、电子工程、智能科学与技术等专业博士研究生、硕士研究生和高年级本科生使用,也可供相关领域的工程技术与科学研究工作者参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

混合免疫智能算法理论及应用 / 刘朝华著. —北京: 电子工业出版社, 2014.1

ISBN 978-7-121-21923-8

I. ①混… II. ①刘… III. ①免疫学—应用—人工智能—算法理论 IV. ①TP183

中国版本图书馆 CIP 数据核字(2013)第 276263 号

责任编辑: 董亚峰 特约编辑: 王 纲

印 刷: 三河市鑫金马印装有限公司

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 720×1 000 1/16 印张: 12.75 字数: 240 千字

印 次: 2014 年 1 月第 1 次印刷

定 价: 38.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

作者简介

刘朝华 1983 年生，湖南衡阳人，工学博士、博士后，现为湖南科技大学信息与电气工程学院讲师，主要从事复杂系统建模、辨识、优化与控制，智能控制理论与应用，计算智能理论与应用，并行计算与云计算，以第一作者在《IEEE Transactions on Cybernetics》(IEEE T-SMC-B)、《自动化学报》、《物理学报》、《电子学报》、《电工技术学报》、《控制理论与应用》、《控制与决策》、《ICIEA》等国际、国内权威学术期刊及重要学术会议上发表科研论文 15 余篇，其中被国际权威检索机构 SCI/EI 收录 12 余篇。作为主要研究人员，参与过国家科技支撑计划重点项目、国家自然科学基金项目、高校博士点基金项目等 6 项。作为项目负责人主持过中国博士后基金面上项目（一等资助），湖南省自然科学基金项目，湖南省博士后科研项目（一等资助）等科研项目。2013 年获得国家留学基金委公派留学项目资助。兼任国内外多个主流期刊、国际学术会议论文评审人。



前言

在人类社会和自然界中存在许多有待人们去研究的奥秘，而人类的智能研究是最为重要的，生命现象的奥秘和生物的智能行为也一直被科学家们所关注。生物体是一个非常大的复杂系统，生物为了维持自身个体的非平衡耗散结构，获得了基于自组织化构造信息处理系统的能力。同时，生物系统也可看成一个高级信息处理系统，基于生物系统启发的信息处理方法能为科学与工程技术领域提供各种富有成效的智能技术。生物智能是在生物的遗传、变异、生长以及外部环境的自然选择中产生的。在用进废退、优胜劣汰的进化过程中，适应度高的结构被保存下来，智能水平也随之提高。基于生物进化、细胞免疫、神经网络、群体智能等某些生物机制可导向计算智能，其代表有神经网络、进化计算、群体智能、免疫计算、模拟退火以及模糊系统等。这些方法具有自学习、自组织、自适应的特征和简单、通用、鲁棒性强、适合并行处理的优点，一直是智能科学领域的研究热点。智能计算在求解问题时不依赖梯度信息，特别适用于传统方法难以解决的大规模复杂非线性优化问题，因此已经被广泛应用于诸多领域，如优化问题求解、复杂工业自动化、航空航天工程、机器人领域、电力系统领域、网络及通信领域、物流系统、计算机领域、半导体制造领域和工程设计领域、系统辨识与故障诊断领域等。随着工业生产过程朝着大型化、连续化、综合化的方向迅速发展，生产过程变得越来越复杂，随之而来的各类工程问题具有大规模、强非线性、大延迟、高变量维数、多求解目标、参数时变等特征，这使得变量的存储及工程优化问题的求解都相当困难。这些复杂工程优化问题因为涉及面广、影响大而成为国民经济及工业界中迫切需要解决的课题。

而复杂工业系统的建模、优化与控制又往往需要高效能的算法来辅助，但是现有依靠单一模式的进化方法往往难以满足该方面的性能要求。混合算

法的思想是利用各算法的优势和各种智能算法内部策略来相互取长补短,从而产生更好的优化能力和求解效率。免疫系统是一种具有高度分布性的复杂自适应学习系统,免疫系统为了抵御外来病原体的入侵维持机体自身耗散结构的平衡,保证生物体的生存与发展,蕴藏着强大的信息处理能力,如“自己-非己”识别能力、免疫细胞克隆扩增、免疫选择、免疫网络、免疫细胞更新、免疫抗体多样等机理,还结合了分类器、神经网络、机器推理等学习系统的优点,具有较好的模式识别功能,呈现出良好的分布性、多样性、耐受性、记忆性、自适应性、鲁棒性等特点。从工程应用角度来讲,免疫系统将先天的经验知识与后天的学习相结合,构造出一种基于知识的自适应动态系统。同时免疫系统经历长时间的进化与其他生物智能系统有着一些共同特性,也容易吸收其他智能系统的优势元素来不断完善免疫系统自身的进化。借鉴免疫系统的机理并结合其他优化算法开展混合免疫智能理论与算法研究能为复杂系统问题的求解提供新的思路和有效的途径。充分考虑多种智能方法之间的有机联系及其算法内在的差异性,通过不断地汲取经典最优搜索方法及现代智能启发式算法的优点,各取所长,建立新模式的混合免疫智能计算理论与算法已是现代免疫计算智能与工程研究的发展方向。

本书的内容是作者近年来在混合免疫智能算法理论及应用研究领域所取得科研成果的归纳与总结。全书深入挖掘生物免疫系统中蕴涵的智能学习机制并结合其他智能处理方法的优点,研究了四类混合免疫智能算法理论方法及其相关应用。希望通过本书,为读者提供混合免疫智能算法模型构建、算法设计、理论分析、算法编程和实际应用等新方法与技术。全书共7章,第1章主要从生物免疫系统的起源阐述了人工免疫的发展,陈述了人工免疫系统中典型的免疫智能算法,接着综述了现有的免疫混合算法及其应用研究现状,并分析了现有混合免疫智能算法研究中存在的主要问题;同时就文中需要研究的应用问题阐述了混沌系统控制与永磁同步电机系统参数辨识研究概况。第2~5章着重于算法模型构建、相关算子设计及实验分析。第6、7章则是基于混合免疫智能算法理论的实际应用研究。通过相应算法在组合优化、数值优化、多模态优化问题、高维问题、动态优化问题等经典复杂问题中的应用,验证了算法的有效性,同时在混沌系统自抗扰优化控制问题、永磁同步电机系统多参数智能辨识与状态监测问题实际应用研究中验证了算法具有解决实际工程问题的优势。

本书的研究成果得到了国家科技支撑计划项目(2012BAH09B02)、国家自然科学基金项目(61174140, 61203309, 51374107)、中国博士后基金面上项目(2013M540628, 一等资助)、高校博士点基金项目(20110161110035)、湖南省自然科学基金项目(14JJ3107)等项目资助。作者的研究成果和本书的撰写工作是在湖南大学章兢教授的悉心指导和帮助下完成的,在此谨致以最诚挚的感谢。感谢湖南大学张英杰副教授、吴建辉博士对本书研究工作的建议与帮助,感谢刘侃博士对本书研究工作所提供的建议与帮助,感谢湖南大学蔡立军教授、邓子云博士对本书工作的支持。感谢湖南科技大学信息与电气工程学院周少武教授、吴亮红博士等领导与同事对本书出版的支持与关心,感谢家人的默默支持与理解。正是这些支持为本书的形成奠定了坚实的基础,在此对他们表示衷心的感谢。本书由湖南科技大学学术著作出版基金资助,感谢湖南科技大学学术著作出版基金对本书出版的资助。

由于人工智能理论与应用正在蓬勃发展,该领域中很多理论技术方法还不完善,混合免疫智能算法与理论及其应用研究领域还有许多科学与工程技术问题需要进一步深入研究和探索,完整的免疫计算智能理论体系还有待进一步完善。加之作者的学识水平与研究工作的范围有限,书中难免存在不足之处,敬请专家与读者批评指正。

刘朝华

2013年11月

目 录

第 1 章 绪论	1
1.1 人工免疫系统的研究概况	1
1.1.1 生物免疫系统	2
1.1.2 人工免疫系统	4
1.1.3 人工免疫算法	6
1.2 混合免疫智能算法研究概况	11
1.2.1 典型混合免疫智能计算模型	12
1.2.2 混合免疫智能算法应用研究进展	19
1.3 现有混合免疫智能算法存在的主要问题及解决途径	24
1.3.1 混合免疫智能算法理论研究的缺陷及解决思路	25
1.3.2 混合免疫智能算法应用研究的局限及拓展	26
1.4 背景与意义	29
1.5 内容安排	30
第 2 章 竞争合作型协同免疫进化算法	32
2.1 引言	32
2.2 竞争合作协同免疫进化模型	34
2.2.1 基于生态种群密度的种群竞争操作	35
2.2.2 群体协同合作操作	36
2.3 种群内部免疫优势克隆选择操作	38
2.3.1 抗体局部最优免疫优势算子	38
2.3.2 基于信息熵的种群多样性控制	41
2.3.3 种群内部抗体克隆选择操作	42

2.4	算法流程及其收敛性能分析	43
2.4.1	竞争合作型协同免疫进化算法步骤	43
2.4.2	收敛性分析	44
2.4.3	时间复杂度分析	46
2.5	实验仿真	46
2.5.1	三种不同形式 ICA 算法性能比较	48
2.5.2	CCCICA 与其他免疫克隆选择算法的性能比较	51
2.5.3	CCCICA 与其他智能算法的性能比较	53
2.6	本章小结	54
第3章	免疫双态粒子群算法	56
3.1	引言	56
3.2	粒子群算法优化原理	58
3.3	免疫双态粒子群算法	60
3.3.1	双态粒子群算法机理	60
3.3.2	精英粒子局部增强学习算子	63
3.3.3	粒子免疫优化	63
3.4	IBPSO 算法流程及性能分析	65
3.4.1	算法 IBPSO 流程	65
3.4.2	算法性能及收敛性分析	66
3.5	仿真实验及分析	68
3.5.1	BPSO (双模态粒子群) 比例参数	69
3.5.2	算法精度比较	71
3.5.3	t-test 测试比较	74
3.5.4	高维函数实验	75
3.5.5	多模态函数及多样性实验	77
3.6	本章小结	80
第4章	免疫协同粒子群算法	81
4.1	引言	81
4.2	免疫协同粒子群进化算法	82

4.2.1	免疫协同粒子群进化模型	82
4.2.2	普通种群混合免疫网络粒子群算法	84
4.2.3	记忆库免疫进化	86
4.3	协作操作	88
4.3.1	个体极值小波学习	88
4.3.2	优势抗体迁移与共享	91
4.4	ICPSO 算法流程	91
4.5	实验仿真及分析	91
4.5.1	算法精度比较	94
4.5.2	算法计算复杂度比较	97
4.5.3	算法 t-test 值比较	99
4.5.4	相关参数及其算子分析	100
4.5.5	动态性能测试	101
4.5.6	ICPSO 与 IBPSO 的性能比较	104
4.6	本章小结	105
第 5 章	免疫蚁群自适应融合算法	106
5.1	引言	106
5.2	蚁群算法优化机理	108
5.3	基于统计学习的免疫融合蚁群算法 (ACALA)	109
5.3.1	带混沌扰动算子启发式蚁群算法	109
5.3.2	MAX-MIN 机制	110
5.3.3	免疫操作搜索算子	111
5.3.4	基于统计学习的自适应免疫蚁群融合算法	114
5.3.5	ACALA 算法收敛性分析及实验仿真	115
5.4	免疫双态蚁群融合算法 (BAIA)	122
5.4.1	双态蚁群机理	122
5.4.2	信息素的更新规则	124
5.4.3	BAIA 算法流程	124
5.4.4	BAIA 算法性能分析	125
5.4.5	BAIA 算法测试实验及分析	127
5.5	本章小结	130

第 6 章 基于混合免疫智能算法的混沌系统自抗扰优化控制	131
6.1 引言	131
6.2 自抗扰控制	133
6.3 基于免疫双态粒子群算法的混沌系统自抗扰优化控制	135
6.3.1 非线性混沌系统	135
6.3.2 评价函数的选择	135
6.3.3 混沌系统自抗扰优化控制系统结构图	136
6.3.4 基于 IBPSO 算法的混沌系统 ADRC 优化控制算法流程	136
6.3.5 实验	137
6.4 基于 IBPSO 算法的混沌系统 ADRC-CMAC 并行优化控制	142
6.4.1 CMAC 结构	142
6.4.2 ADRC 与 CMAC 并行控制算法	143
6.4.3 基于 IBPSO 的 ADRC-CMAC 优化设计流程	144
6.4.4 实验	144
6.5 本章小结	150
第 7 章 基于混合免疫智能算法的永磁同步电机系统多参数辨识与状态监测	151
7.1 引言	151
7.2 PMSM 数学模型	153
7.3 基于免疫智能算法的永磁同步电机多参数辨识与温度监测模型	155
7.3.1 基于 ICPSO 的永磁同步电机多参数辨识思想	155
7.3.2 基于 ICPSO 的永磁同步电机多参数辨识与温度监测模型	157
7.4 实验及分析	160
7.4.1 实验方案与平台	160
7.4.2 实验结果	163
7.5 本章小结	173
后记	174
参考文献	177

第 1 章

绪 论

1.1 人工免疫系统的研究概况

自古以来人们就不断对生物系统及人类自身功能结构进行模仿并从中提取有用的信息处理方法。例如，人工神经网络是对人脑结构感知的模拟^[1]，模糊优化与控制思想模拟了人类模糊逻辑推理^[2,3]，遗传进化算法仿效了大自然的优胜劣汰自然选择过程^[4]，蚁群算法模拟了蚂蚁群体社会信息素传递机制^[5,6]，粒子群算法仿效了鸟类、鱼类等的微粒觅食及迁徙行为^[7]。以上的生物信息行为模拟属于仿生智能范畴。对仿生智能算法的研究和应用业已成为智能系统领域的研究热点，人们相继提出了诸多求解复杂问题的智能处理方法，其中具有代表性的方法有人工神经网络、模糊系统、进化计算、多 Agent 系统、蚁群算法、粒子群算法、DNA 计算、免疫计算等^[1~8]。近年来，随着生物免疫系统的深入发展，免疫计算智能开始进入人们的视野，已成为人工智能系统领域研究的一个新热点。自 1974 年，丹麦学者 Jerne 提出了独特性免疫网络模型^[9]，开启了人类对人工免疫系统研究的序幕。人工免疫系统^[8~11]是模拟生物免疫系统中的智能信息处理机制，且具有高度并行化的分布式、自组织、自适应等特征的智能系统。人工免疫系统的研究成果涉及数值优化^[12~15]、组合优化^[4,7,16~17]、多目标优化^[5,18~20]、故障诊断^[1,21~23]、计算机系统安全^[10,24~26]、系统辨识^[27~30]、智能控制^[31~35]、复杂系统优化与控制^[36~37]、数据挖掘^[2,38~41]及图像处理^[42~45]等诸多领域。随着人们对人工免疫系统的深

入研究，一方面不断地挖掘生物免疫系统蕴涵的智能信息处理机制并构造新的免疫优化机理；另一方面不断融合其他算法的优点，优势互补，而演化出一些新的混合免疫智能优化理论与技术。

1.1.1 生物免疫系统

生物免疫系统^[46]是生物机体的一种对外界环境特异性的生理反应，能够自动识别自身与清除抗原性入侵来维持自身非平衡耗散结构组织体的稳定；同时免疫系统是生物机体执行免疫功能的机构，是产生免疫应答的物质基础。免疫系统对自身细胞发生突变（如细胞病变、细胞癌变等）以及侵入机体的非己成分（如病毒等各种病菌体）具有精确识别、自适应应答及清除等功能。现代生物免疫系统的研究重点在于免疫动态响应：抗原、免疫细胞及其分泌的抗体、细胞因子等机体相互作用的动态过程。在生物体内存在的 B 细胞和 T 细胞是产生免疫响应的重要免疫细胞，它们都有着各自的分工。B 细胞的功能是产生抗体，且每个 B 细胞只能产生一种抗体，通过抗体与抗原的匹配，实现对抗原的清除；与此同时，T 细胞在免疫响应过程中能刺激和抑制 B 细胞的增殖与分化，对免疫调节有着积极的作用。生物免疫系统的研究发展历程归纳如下。

早在 17 世纪，中国民间医学家就发明了以人痘来预防天花的事例。接着英国医生 Edward Jenner 在 1796 年从“牛痘”中提取疫苗，标志着现代医学免疫的开端。而后来法国免疫学家 Pasteur 创造了减毒细菌疫苗并兴起了主动免疫的方法，奠定了经典免疫疫苗的基础。经过数百年的发展，免疫学逐渐从微生物学中的一小门类发展成为一门独立的学科，并派生出若干分支，形成了以细胞免疫学、分子免疫学、神经与内分泌系统免疫学、行为免疫学等基础免疫学研究体系。免疫学史上（20 世纪 90 年代初）比较重要的思想、理论和研究成果^[8,47,48]如表 1.1 所示。

表 1.1 生物免疫学的主要理论研究成果

发展简史	时 间	代表人物	思想、理论和研究成果
经验免疫学时期	17 世纪起	中国医学家	“人痘” 苗的发明和推广应用
		Jenner	“牛痘” 苗的发明和推广应用
	1796—1870 年	Koch	病理学
		Pasteur	减毒疫苗接种
		Besedovsky	神经—内分泌—免疫网络学说
		Metchinikoff	噬菌作用
科学免疫时期	1910—1930 年	Bordet	免疫特性
		Ehrlich	发现细胞受体
	1930—1950 年	Breinl&Haurowitz	抗体合成
		Landsteiner	半抗原结合
现代免疫时期	1930—1950 年	Breinl&Haurowitz	抗体合成
		Linus Pauling	抗原模型
	1950—1980 年	Burnet	克隆选择学说
		Niels Jerne	免疫网络理论学说
分子水平研究	1980 年至现在	Susumu Tonegawa	受体的结构与多样性理论

生物免疫系统智能特征如下。

① 免疫系统是由若干细胞和分子组成的生物体，是一种没有中心控制器的分布式系统，同时是一类能有效地处理复杂问题的非线性自适应网络。由于免疫系统存在多种不同的 B 细胞，且 B 细胞群体之间的相互反应能在动态变化的环境中维持其个体平衡。

② 免疫系统是一个自组织记忆存储器，它具有记忆与遗忘功能，通过免疫应答对外界进行无导师自适应学习，能够自适应外界环境变化。

③ 免疫系统中存在不同抗原与抗体的匹配及基因重组现象，保证了免疫系统的多样性；免疫系统中抗体多样性、克隆复制、亲和度变异、免疫性选择、基因重组等机理，可用于复杂问题的求解与寻优。

④ 借鉴独特型网络、互连耦合免疫网络、免疫反应网络 and 对称网络免疫网络学说能建立人工免疫网络模型，且免疫系统中 B 细胞和 T 细胞之间的相互反应维持系统动态平衡可以增强免疫算法的多样性。

⑤ 自身抗体对外界抗原的快速反应及快速维持免疫系统稳定的免疫反

馈机制可用于建立有效的反馈控制系统,如基于 T 细胞的免疫反馈原理可用于设计自调整免疫反馈控制器。

⑥ 免疫系统存在的免疫耐受诱导及维持机理,该机理允许抗原被相同的抗体识别且能对抗原中的噪声进行耐受,其机理可用于发展系统故障诊断方法。

⑦ 免疫系统的自体集与非我集的识别,可以清除外来抗原的入侵,借鉴此机理可以应用于计算机系统安全及入侵检测。

1.1.2 人工免疫系统

1958 年, Burnet 等人提出了生物克隆选择学说^[49],该学说表明生物免疫系统存在抗体-抗原自适应刺激的动态过程现象,在这一生命现象中所体现出的学习、记忆、抗体多样性等生物智能特征,正是建立人工免疫系统的基础。20 世纪 70 年代,美国著名免疫学家 Jerne 提出了免疫系统的网络学说,基于该学说并建立免疫网络的一般数学模型^[50],开创了人工免疫系统的研究。而后有关人工免疫系统研究的报道不断出现,如 Perelson 对独特型网络进行了进一步深入的研究与探索^[51]。1986 年, Farmer 进一步诠释了基于免疫网络的假说,构造出一种免疫系统的动态数学模型,并与 Holland 的分类器系统进行对比研究,提出了一系列有价值的智能信息学习方法。他们的研究作为建立有效的基于免疫机理的人工智能系统和计算智能系统及其发展奠定了坚实的基础^[52]。根据 Burnet 的抗体克隆选择学说, De Castro 与 Kim 等人从不同的角度模拟了上述生物学抗体克隆选择实现过程,相继提出了一系列不同的免疫克隆选择算法,并建立了克隆选择算法基本框架,将其应用于复杂优化问题求解^[53,54]。20 世纪 90 年代初期,人们开始将免疫系统原理应用于工程问题求解,有关免疫机制用于解决实际工程问题的研究开始逐渐报道^[55]。1996 年 12 月,日本首次举行了基于免疫系统的国际专题讨论会,最先提出了“人工免疫系统”(Artificial Immune System, AIS)的概念。随后,人工免疫系统进入了发展期,人工免疫系统作为计算智能的崭新分支开始成为诸多国际期刊及国际学术会议的重要议题。1997 年开始,人工智能领域著名国际会议 IEEE System, Man and Cybernetics 每年均组织专门的 AIS 研讨会(Workshop);国际进化计算顶级杂志 IEEE Transaction on Evolutionary

Computation 在 2001 年和 2002 年还相继出版了有关 AIS 的专辑；同时人工免疫系统研究主题也受到人工智能其他著名国际会议 International Joint Conference on Artificial Intelligence (IJCAI)、International Joint Conference on Neural Networks (IJCNN)、GECCO (Genetic and Evolutionary Computation Conference)、CEC (Congress on Evolutionary Computation) 等的关注；2002 年 9 月，在英国 Kent 大学召开了第一届人工免疫系统国际学术会议 ICARIS (1st International Conference on Artificial Immune Systems)。自此，ICARIS 主要在英国、意大利、加拿大等国家每年相继轮回召开，有关人工免疫系统的主题研究开始从单一的免疫系统理论研究走向多方位应用研究。在我国，西南交通大学的靳蕃教授等人于 20 世纪 90 年代初在所著《神经网络与神经计算机原理应用》一书中就已经指出“免疫系统所具有的信息处理与肌体防卫功能，从工程角度来看，具有非常深远的意义”^[56]，开启了国内有关免疫系统信息提取机制的萌芽。从 1997 年开始，国内逐渐出现有关免疫计算的研究报道，近年来得到快速发展。2002 年国内学者莫宏伟出版了国内第一本介绍人工免疫系统的著作《人工免疫系统原理及其应用》^[57]，该著作阐述了从免疫系统生物学原理到人工免疫系统领域的研究成果及其相关工程技术应用。西安电子科技大学焦李成教授所领导的团队相继研究了一系列人工免疫优化方法，如免疫遗传算法、免疫进化规划、免疫进化策略、量子免疫克隆算法等，并证明了算法的收敛性，分析了算法的复杂度以及相关参数效应，无论在理论方法还是算法设计上均有所创新，构建了免疫克隆计算智能的基本框架，为免疫计算智能理论与应用的进一步研究提供了理论指导和技术支持，并编著了免疫优化方面的系统理论著作《免疫优化计算-学习与识别》^[8]。肖人彬等人将人工免疫系统与工程领域结合研究并阐述了工程免疫计算的基本理论、集成方式、求解算法及技术实现^[58]。最近还有学者开始将人工免疫系统研究纳入自然计算范畴以拓宽人工免疫系统研究范围，便于免疫系统研究成果得到更广泛工程应用研究，期望促进人工免疫系统继续深入发展^[59]。随着人工免疫系统的发展，人们开始将目光投向免疫系统机理与其他智能算法进行结合展开混合免疫计算智能理论与方法及其相关工程技术应用研究。

1.1.3 人工免疫算法

生物免疫系统也可看成一个高级信息处理系统，基于生物免疫系统启发的信息处理方法能为工程技术领域提供各种富有成效的智能技术。典型的人工免疫算法有一般免疫算法、克隆选择算法、免疫网络算法、阴性选择算法。

1. 一般免疫算法

受生物免疫系统机制启发，通过仿效人体的免疫系统，结合体细胞和免疫网络理论，以实现生物免疫系统的抗原自动识别、细胞复制与分化、免疫记忆和自我调节功能的模拟。基本免疫算法流程如图 1.1 所示。

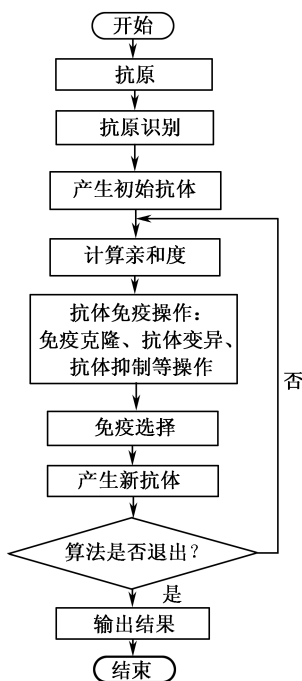


图 1.1 基本免疫算法流程

免疫算法一般框架模型

步骤 1: 初始化种群, $t=0$; $A(0)=[a_1(0) a_2(0) \cdots a_n(0)] \in I^n$, 式中 I 为抗体空间, t 为算法执行代数, A 为抗体集合;

步骤 2: 计算亲和度, $A(0):[\Phi(A(0))]=[\Phi(a_1(0)) \Phi(a_2(0)) \cdots \Phi(a_n(0))]$;

do{ 进行抗体免疫相关操作:

抗体克隆: $A'(k)=T_c^C(A(k))=[T_c^C a_1(k) T_c^C a_2(k) \cdots T_c^C a_n(k)]^T$;

抗体变异: $A''(k)=T_m^C(A'(k))$;

抗体抑制: $A''(k)=T_i^C(A''(k))$;

计算亲和度: $A'''(k):[\Phi A'''(k)]$;

抗体选择: $A(k+1)=T_s^C(A'''(k))$;

得到新抗体:

$$A(k+1):[\Phi(A(k+1))]=[\Phi(a_1(k+1)) \Phi(a_2(k+1)) \cdots \Phi(a_n(k+1))];$$

$$T=t+1$$

}while(算法终止条件判断是否为真);

每个具体的人工免疫算法的基本操作过程都可以以上述一般免疫算法框架为基础, 根据每种具体免疫算法的特点, 通过对上述一般算法模型做适当修改而成。

2. 克隆选择算法

1958 年, 免疫学领域的著名学者 Burnet 等人根据生物免疫系统原理最先提出了克隆选择学说^[46], 该学说核心思想为: 抗体作为天然产物, 以受体的方式遍布于细胞表面, 而抗原具有选择性地与相对应的抗体进行反应。抗原与相应抗体受体之间的相互刺激等反应可导致细胞克隆增殖与快速分化。在具有相同的抗体特异性细胞群体中, 一部分细胞克隆被分化为抗体生成细胞, 另一部分则形成免疫记忆细胞参加接下来的免疫应答操作。在此过程中, 借助克隆操作使其激活、分化与增殖, 以实现抗体群规模的增大, 再通过免疫应答反应以达到体内抗原最终清除。因此, 克隆选择是生物免疫系统内自适应性抗原刺激的动态过程, 且在这一过程中体现出学习、记忆、抗体多样性等生物智能特征, 而这些生物智能特征恰是建立人工免

疫计算智能模型的理论依据。根据 Burnet 的抗体克隆选择学说，De Castro 提出一种经典通用的一般克隆选择算法框架^[11]，基本克隆选择算法流程如图 1.2 所示。

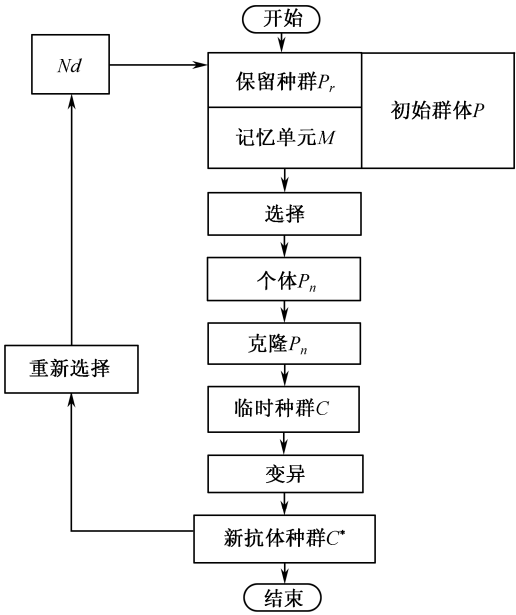


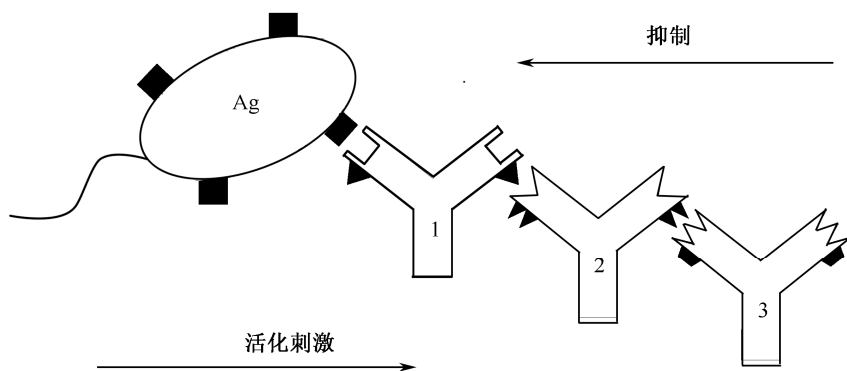
图 1.2 基本克隆选择算法流程

基本克隆选择算法描述

- 步骤 1：首先生成候选解 P ， P 由记忆单元 (M) 和保留种群 (P_r) 组成，即 $P=P_r+M$ 。
- 步骤 2：根据亲和度计算，选择 n 个较好个体 (P_n)。
- 步骤 3：对种群中 n 个较好的个体进行克隆复制，生成一个临时抗体克隆种群 (C)，抗体克隆规模与抗体-抗原的亲和度成正比。
- 步骤 4：对临时克隆种群中的抗体进行高频变异，这里高频变异与抗体-抗原的亲和度相关联，获得一个经过变异后的抗体群 (C^*)。
- 步骤 5：从 C^* 中重新选择亲和度得到提升的个体组成记忆单元 M 。 P 中的一些个体也被 C^* 中其他亲和度高的个体所替代。
- 步骤 6：利用新产生的抗体代替 d 个亲和度低的旧抗体 (引入多样性保持机制)。

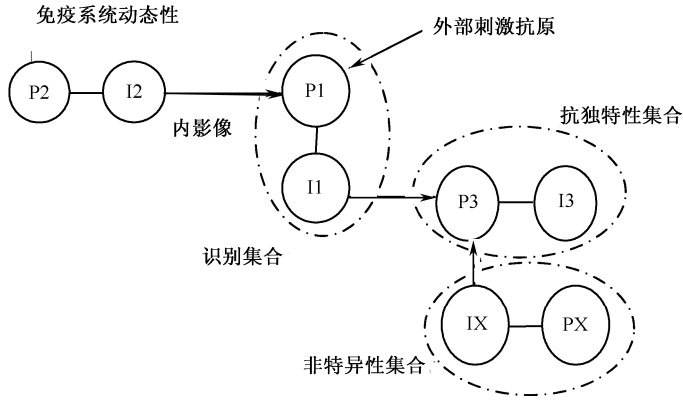
3. 免疫网络算法

从生物免疫网络学说中得到启发，人们开始进行人工免疫网络研究，目前对免疫网络理论研究已有多种模型，主要包括独特型免疫网络模型、免疫反应网络模型、互连耦合免疫网络模型、对称网络和多值免疫网络模型等。其中，最具有影响力的是独特型免疫网络模型，如图 1.3 所示。该模型是最初由 Jerne 在 1974 年根据现代免疫学原理对抗体分子特异性的认识并结合 Burnet 的克隆选择学说所提出的一类网络模型^[9]，独特型网络理论认为在抗原的表位 Epitope 与抗体的对位 Paratope 之间存在相互刺激和抑制关联，还表明抗体本身也具有抗原特性，在其表面存在表征抗原特性的独特位 Idiotope，抗体不仅能识别相应的抗原，而且还能识别体内其他抗体，这样便形成了抗原-抗体、抗体-抗体之间相互作用的动态网络体系结构。这种免疫网络结构表明免疫细胞具有动态特性，不仅保持了机体内适当的免疫强度及其内部环境的稳定，而且还能维持抗体群的多样性。



(a) 免疫系统动力学

图 1.3 独特型免疫网络模型



(b) 抗原内影像

图 1.3 独特型免疫网络模型 (续)

根据独特型免疫调节网络理论, De Castro 构造了一种模拟免疫网络对抗原刺激的应答过程的 aiNet 算法^[17], 成功地模拟了抗原-抗体、抗体-抗体之间的动态稳定平衡、抗体学习、抗体群的多样性保持等过程, 其中 aiNet 网络算法学习过程描述如下。

-
- 步骤 1: 随机初始化抗体群, 选择 n 个高亲和力抗体, 复制 (克隆) 种群中 n 个最好的个体, 生成一个克隆临时种群 (C)。
- 步骤 2: 对克隆临时种群进行高频变异, 这里高频变异和抗体、抗原的亲度相对应, 由此获得了一个变异后的抗体群 (C^*)。
- 其中 $C^* = C - \alpha \cdot (C - X)$, C 是网络细胞群, X 是抗原细胞, α 是学习率或变异率, 变异使抗体朝向识别抗原的方向进化, α 的取值依据网络单元与抗原的亲度而定, 亲度越高则 α 的取值越小。
- 步骤 3: 从 C^* 中选出一定比例具有高亲和力的抗体, 放入克隆记忆库集中。
- 步骤 4: 对记忆抗体相似度进行计算, 进行网络抑制操作。
- 步骤 5: 免疫网络抗体生成。
- 步骤 6: 终止条件判断, 抗体达到设定条件或指定迭代次数完成。
-

4. 阴性选择算法

阴性选择原理源于生物免疫系统对外界抗原的特异性反应, 仿效生物免疫系统中胸腺 T 细胞产生过程中的“否定选择”现象。根据这一原理 Forrest

最先研究了一种基于阴性选择原理的异常数据变化检测方法^[24]，自此阴性选择机制开始应用于计算机系统安全领域。阴性选择过程其实际是通过识别“自体集”与“非体集”的信息，通过随机产生检测器，而后删除那些检测到自我的检测器，最后保留那些检测到“非己”的检测器。其算法模型如图 1.4 所示。

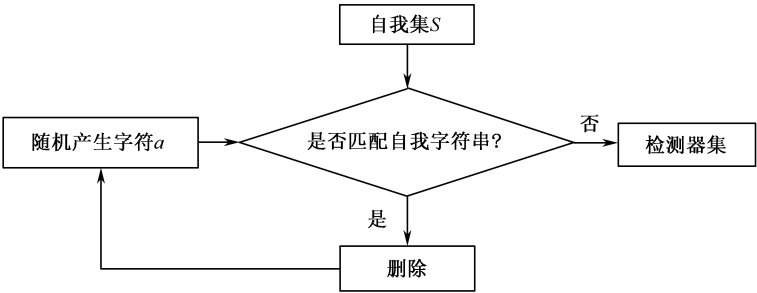


图 1.4 阴性选择算法模型

阴性选择流程描述

- 步骤 1：定义一长度为 L 的字符串自体集合 S 。
- 步骤 2：随机产生一串长度为 L 的字符向量 a 。
- 步骤 3：将字符向量 a 依次与集合 S 中的字符向量进行匹配。
- 步骤 4：依据相关匹配规则，如果 a 遇到集合 S 中相匹配的字符向量，则结束本次字符匹配，转到步骤 2 重新匹配。
- 步骤 5：如果 a 不与集合 S 中任何字符向量相匹配，则向量 a 成熟，将 a 加入检测器集合中。

1.2 混合免疫智能算法研究概况

No free Lunch 定理^[60]已经证明，没有一种方法对所有问题都是有效的，各种方法有其相应的优势，也有其相应的劣势。特别是对于一些不确定和非线性的复杂系统问题的求解，单一模式优化方法的局限性难以获得最优解或满意解。混合算法的思想是利用各种智能优化算法内部独特的优点和机制来相互取长补短，通过相互交叉与融合所演化的混合智能系统能产生更好的优

化性能和求解效率。因此混合智能优化方法可以兼顾问题的各个方面,能为复杂系统优化与控制提供有效的解决方案。混合优化算法一般由以下三种混合方式组成。

① 算子移植/内嵌法:该方法通过将算法 A 的优秀算子移植/内嵌到算法 B 中,从而弥补算法 B 的缺陷。

② 双模切换法:该类方法利用不同算法各自的优点,通过切换当前所用算法种类来克服寻优过程中出现的误收敛、难收敛等问题,即算法 A 搜索到一个较优的可行解后,立即切换到算法 B 进行求解。

③ 共同求解法:两种或多种算法相结合对一个问题进行共同求解,集成各种算法的优势,其中一种方法作为算法的主框架,其他方法作为主框架中的必要组成部分,各种方法共同协作搜索。

1.2.1 典型混合免疫智能计算模型

1. 免疫遗传算法

遗传算法是一种仿效自然界的优胜劣汰自然选择过程的随机搜索算法,其机制中存在交叉、变异、复制、重组等操作。现有研究表明经典遗传算法在寻优及其应用中,表现出一定的局限性,如搜索接近全局最优解时,其寻优速度减慢,易陷入局部极值点。利用免疫系统的抗体多样性机理及免疫选择机理可以克服遗传交叉与变异操作中的退化现象,以提高遗传算法寻优能力。同时免疫系统吸收遗传系统中交叉机制同样也可以加快免疫算法的收敛性。Forrest 等人最先提出基于免疫机理的遗传算法建模方法^[24],开创了免疫遗传算法相关研究。西安电子科技大学焦李成教授构造了基于免疫系统的遗传算法一般优化模型^[4],并系统阐述了算法模型及免疫算子构造机理。有不少研究者对免疫遗传算法个体选择概率进行了探索,为避免传统基于精英选择策略的缺陷,如陈曦等人在文献[61]中提出了运用适应度与浓度相结合的选择方式来保证群体多样性,增强学习能力。Jiao 等人在文献[4]中将退火个体选择概率思想融入免疫遗传算法中,构造了基于退火原理的免疫选择方法以接受次优解,避免抗体群被局部极值点充满,保证算法的全局寻优能力。还有一些学者将免疫算法与多种群遗传算法相结合,如杨东勇等人在文献[62]

中根据免疫系统形态学空间覆盖原理及特征信息,将自体集划分为多个子群体,利用遗传算法对各个划分子种群进化优化并得到成熟检测器。该方法有效地降低了检测器的冗余度,提高了数据检测效率。葛宏伟等人在文献[63]中,引入免疫系统中的二次免疫应答、免疫记忆、免疫选择、浓度控制等机理构造出一种更接近实际免疫系统的免疫遗传算法。研究人员还对免疫遗传算法的收敛性进行了理论证明,免疫克隆选择是通过亲和度导向的抗体随机映射进化过程,抗体群的状态转移情况是满足 Markov 随机过程并以概率 1 收敛^[4]。

2. 免疫粒子群算法

粒子群优化算法^[73]是由 Kennedy 等人于 1995 年提出的一种模拟鸟类、鱼群等的微粒觅食行为的进化计算方法,该算法通过群体中个体之间的邻域信息协作来进行迭代优化。大量研究表明粒子群算法在进化后期容易出现进化停滞现象。于是一些学者将免疫系统原理引入粒子群算法中以增强粒子群多样性,促使微粒具有大范围变异能力,加快收敛速度。Ge 等人在文献[7]中提出了一种先经粒子群优化然后再经免疫克隆选择优化的串行混合智能优化算法结构,将人工免疫系统中的疫苗接种和受体编辑机理引入粒子群算法中,提出了一种免疫粒子群优化算法模型。该算法在执行过程中每隔数代就对群体中部分微粒进行疫苗接种以加快算法收敛速度,通过受体编辑重组抗体结构增强群体多样性。陶新民等人在文献[64]中利用免疫阴性算法对问题空间进行多群体划分,通过种群进化信息生成优胜解区域,来指导变异的粒子群向最优解子空间逼近,提高了粒子群算法逃出局部最优的能力。

文献[13]提出了一种免疫粒子群网络算法,该算法基于粒子群优化特性和免疫网络理论各自优势的基础上,综合利用粒子群局部搜索能力、用免疫动态网络抑制策略等优势,保持种群的多样性;对粒子群的规模进行自适应调节,提高了算法对复杂的问题的求解效率。Lu Hong 等在文献[65]中针对粒子群算法多样性差,易陷入局部最优且收敛速度慢等缺陷,提出了一种基于免疫机制的粒子群优化算法,该算法结合粒子群算法的全局寻优能力和免疫多样性保持机制,改善了粒子群算法摆脱局部极值点的能力,提高了算法的收敛速度。对典型的多模态函数优化结果表明,该算法能有效地抑制算法

早熟。文献[66]综合了粒子群优化算法简单快速的特点和免疫系统种群多样性保持好的特性,提出了一种基于免疫系统信息处理机制的二进制粒子群优化算法,改善了进化后期算法的收敛性能和全局寻优能力。焦李成等人在文献[67]中抗体样本采用正交初始化编码来保证群体初始化分布均匀性,融合免疫算法多样性与粒子群算法快收敛性的优点,提出了一种免疫克隆粒子群多目标优化算法。文献[68]充分利用粒子群的协作型及免疫网络的动态抑制过程,提出了基于粒子群协作型人工免疫网络模型,该算法在运行过程中通过增加全局粒子群协作算子,使其人工免疫网络中的记忆细胞具有粒子的特性,同时采用基于可变步长变异的克隆选择算子使算法具有高精度搜索能力。李中华等人在文献[69]中借鉴免疫网络及粒子群算法各自的优点提出了一种新型人工免疫网络优化算法。

3. 免疫-蚁群混合算法

蚁群算法是 Dorigo 等人提出的一种模拟蚂蚁群体在觅食过程中依赖信息素通信的群居智能行为的仿生进化算法。蚁群算法具有分布式并行全局搜索能力,通过信息素的调节收敛于最优目标值,大量研究表明蚁群算法运行前期收敛速度快,但进化后期出现停滞现象,且易陷入局部极值点。借鉴免疫系统的多样性、超变异、记忆等机理可以指导蚁群信息素变异,从而改善蚁群算法求解能力。Ashish 等人提出了一种混合免疫蚁群智能算法^[5],通过引入免疫算法多样性及超变异能力等特征来指导蚁群信息素生成与更新,通过免疫算法操作后的信息素进一步指导蚁群算法寻优,提高了算法搜索能力。Ling Q 等人在文献[70]中通过模拟生物免疫系统来保持蚁群算法的多样性,该算法采用免疫选择、免疫记忆、免疫代谢、密度控制等机制,实验结果表明该算法收敛速度快,具有较强的寻优速度,能有效抑制算法早熟。焦李成等人在文献[14]中利用蚁群信息素的无导师自学习机制并结合免疫克隆选择机制,构造出一种基于信息素模因机理的免疫克隆选择算法,该算法无须提供确定候选的局部搜索策略,降低算法局部搜索代价,将免疫克隆选择算法与信息素学习机理相结合,可以借鉴抗体种群的进化经验来提升算法的寻优效率。

4. 协同人工免疫算法

协同进化是一种新型的进化算法框架^[71],它借鉴了自然界中不同物种的协同进化机理,是传统的单物种独立进化机制的发展,考虑了自然生态环境对种群进化的影响,多个种群间相互作用共同进化,从而推进整个系统不断演化。近年来,人们试图将协同进化与人工免疫系统进行结合研究,期望提高人工免疫算法对复杂问题的求解性能。文献[72]考虑了自然环境对生物进化的影响,在免疫算法与遗传算法基础上,提出了一种考虑资源环境作用的协同进化免疫遗传算法,同时设计了克隆环境演化算子及自适应搜索算子,算法采用多种群协同进化方式,提高了抗体群的寻优效率。王磊等人文献[73]中借鉴了自然进化中协同进化的思想来改善当前免疫计算模型中的一些缺陷与不足,提出了一种疫苗与种群同步进化,变异算子也同步进化的协同人工免疫智能计算模型,实验结果表明提高了免疫算法全局搜索效率。文献[74]结合自然界协同进化理论、免疫系统、量子计算理论,提出了一种量子协同免疫智能计算模型,该算法中抗体基于量子比特编码,通过量子旋转门变异及量子策略来更新个体,各种群进行独立进化,充分考虑了量子间的协作与竞争,求解过程中利用群体中动态信息来指导种群进化。Luiz 等人文献[75]中建立一个协同免疫算法模型,该算法根据需要将群体分为两部分,一部分负责规则学习,另一部分负责数据优化,同时运用基于梯度下降变异策略的人工免疫优化算法对上述两部分不同群体进行优化,两群体进行合作协同进化,促进信息交流加快亲和度成熟。Henry 等人文献[33]中将生物免疫识别机制与 Agent 协作型协同演化思想相结合,提出了一种免疫协同多 Agent 系统模型,提高了算法对复杂问题的求解能力。

5. 混沌免疫算法

混沌优化算法是一种基于非线性系统的新型搜索算法,其核心思想是把待求问题的变量从解空间映射到混沌空间,然后再利用混沌变量所具有的随机性、规律性、遍历性等特点对解区间进行搜索求解。利用混沌变量对初值的敏感性,若干不同的初值向量可以得到相应的混沌变量。混沌优化步骤主

要包括混沌编码、混沌初始化、混沌遍历搜索等。

混沌对初值敏感,易跳出局部极小点且搜索速度快,研究表明当搜索空间比较大时其效果并不显著,而免疫进化算法具有迅速将问题的解限定于一个较小区间内的优点。综合免疫进化算法全局搜索能力与混沌优化算法局部搜索能力强的优势,一些新的混沌免疫算法得到报道。文献[76]利用混沌变量的遍历性对初始抗体群进行粗粒度搜索,然后再通过免疫克隆选择算子、变异算子、抗体的循环补充算子等精细局部寻优操作,避免了群体陷入局部极值并保证了算法向全局最优值逼近。张海英等人提出了一种混沌免疫进化算法^[77],该算法首先利用混沌序列初始群体;其次对群体中的个体亲和度进行变换以调节个体的选择概率;最后利用概率分析方法,并证明算法具有全局收敛性。文献[78]结合混沌优化与免疫系统的特征,采用折叠次数无限的自映射 $x=\sin(2/x)$ 模式混沌变量来设计自适应变尺度混沌免疫优化算法,将自适应变尺度引入算法中以自适应调整优化变量的搜索空间,该文所提方法在局部搜索能力和搜索精度上均优于传统免疫进化算法,克服了混沌优化方法在变量大范围内失效的缺点,保证了算法全局收敛性。杨海东等人在文献[79]中将混沌理论和克隆选择原理相结合,建立新的自适应混沌超变异优化方法以增强算法的局部搜索能力;其次算法从基因型和表现型两方面构造出基于抗体激励水平的免疫网络调节机制,保持了抗体群的多样性。

6. 量子免疫算法

在量子计算理论中,各个状态间的迁移是通过量子门的旋转变换矩阵来实现的^[18],通过量子旋转门的旋转角度来表征变异操作。量子智能信息处理主要包括量子编码、量子变异算子、量子交叉等步骤。近年来有关量子计算与人工免疫系统结合研究取得了一些成果,如 Jiao 等人在文献[18]中结合量子信息处理特征与人工免疫系统原理,提出了量子免疫克隆选择算法,通过量子变异与重组来加速抗体成熟,并将该算法用于多目标问题求解,提高了算法对复杂问题的全局优化能力。文献[80]将量子信息与免疫克隆选择算法进行结合研究,算法充分结合了免疫克隆算子局部寻优能力及量子交叉信息传递的优势。算法采用量子位对抗体群进行编码以增强抗体群多样性,通过量子旋转门的动态调整旋转角度策略对抗体进行演化与更新加速了克隆算

子收敛速度；利用免疫克隆操作的局部寻优特性在各个子群体内部运用量子交叉操作以增强种群优良信息交流，实验结果表明该算法兼顾了种群的多样性和抑制了算法早熟。马文萍等人在文献[43]中将量子交叉操作引入人工免疫系统中，对同一抗体进行多方向克隆选择操作，各个子群体间采用量子交叉算子以增强抗体间的信息交互，有效地克服了算法早熟。文献[81]中将免疫克隆选择理论与遗传量子算法进行结合研究，提出了免疫克隆量子算法(CQA)，在算法中抗体基于量子编码，量子个体和观测态之间互动进化，加快算法 CQA 的收敛速度并减少算法的计算复杂度，较好地解决了多用户检测问题。

7. 模糊免疫系统

模糊逻辑^[2,3]模拟复杂的生物过程中具有的不确定性和自适应性，具有较好的环境自适应性。研究人员试图将模糊逻辑与人工免疫系统进行结合研究。文献[82]提出了一种基于模糊聚类和遗传算法相结合免疫算法阴性选择的异常数据检测模型，利用模糊聚类和遗传算法进行混合以形成基于负选择算法的免疫检测模型，其识别率与类似算法相比较有明显的提高。文献[44]提出了模糊免疫网络算法，算法首先利用免疫网络理论来构建免疫网络算法对已知样本进行非监督学习，动态地确定已知样本的聚类数及聚类中心，然后利用模糊聚类对所获结果进行逐一归类。最近有些研究人员通过引入模糊规则自适应地调整免疫优化算法中的变异概率，通过向种群中注入新基因，以防止优秀基因片段过度繁殖而陷入局部最优点。例如，文献[83]对生物免疫机理和模糊逻辑原理进行了结合研究，设计了一类新颖的模糊自适应免疫算法，通过采用混沌超变异操作来增强算法搜索能力，引入模糊逻辑来调节算法参数，提高了算法的自适应能力，并借助免疫网络调节策略保持了抗体群的多样性，提高了算法全局寻优能力。文献[84]在模糊相似度及背离度的基础上提出了基于变阈值可变半径的免疫阴性选择算法构造有效检测器集，通过调整匹配阈值半径来降低黑洞数量，以获得最佳检测器集合，有效避免了检测器集中数据冗余现象。近年来还有研究人员借鉴人工免疫算法的强优化能力，对模糊控制器进行优化设计，如对控制器参数、模糊规则表、模糊子集隶属函数的参数进行优化。文献[85]在模糊控制规则确定的条件下，提

出了采用免疫算法优化模糊控制器各参数,并将基于免疫优化设计的模糊控制器应用于汽车主动悬架控制。Liao 等人在文献[34]中将免疫粒子群算法应用于模糊神经网络优化,并将其用于火炉温度系统解耦控制。随着智能控制理论的发展,相关研究人员将免疫反馈控制与模糊控制器结合构建复合控制策略。如文献[86]将生物免疫反馈机理与模糊控制规则相结合构造一种复合控制器,该控制算法由一个常规的 PID 控制器和一个免疫型比例控制器顺序串联组成,其中免疫比例控制器的非线性函数由模糊推理实现,最后将该控制器应用于永磁交流伺服系统之中。文献[87]提出了一种新颖的基于生物免疫系统反馈机理的通用控制器结构。

8. 免疫系统与神经网络混合

神经网络(Neural Network, NN)是一类复杂非线性动力学系统,具有自组织、自学习、鲁棒性强等优点,可作为联想记忆和优化计算工具。将免疫算法与人工神经网络相结合,可以改善人工神经网络的性能,也可以利用免疫算法来设计神经网络或者对神经网络的拓扑结构和权值进行优化学习。在神经进化计算中,抗体编码代表神经网络的参数(包括权值、阈值和连接度等),依据免疫选择的原理对其进行变异操作和抗体重组。曹先彬在文献[88]中运用一种免疫进化规划来设计多层前馈神经网络,将生物免疫中浓度调节机制与多样性保持机制引入传统进化规划中构建新的免疫进化规划算法,然后利用免疫进化规划算法来设计多层前馈神经网络。Muhammad 在文献[2]中设计了一种基于免疫系统的模糊神经网络识别系统,对物体气味进行分类识别。文献[89]中利用了人工免疫系统的记忆、学习和自组织调节原理,提出了一种基于免疫原理的新型径向基函数(RBF)神经网络模型,通过对 RBF 神经网络隐层中心向量和位置进行选择优化,并结合递推最小二乘算法来确定网络输出层的权值,最后将构造的新型免疫 RBF 神经网络方法应用于热工过程的非线性建模。文献[90]在分析子波神经网络内在机理的基础上,提出了采用免疫进化算法对子波网络进行学习训练。充分利用免疫进化算法的并行运算规则、智能搜索方式及概率判断准则,这样便弱化了子波网络的应用条件,加强了联想记忆和信息处理的能力。文献[91]基于适应性免疫应答

中的优化机理,提出了一种免疫进化算法用于 RBF 模糊神经网络控制器参数优化设计的方法。侯家利等人在文献[92]中讨论了免疫网络和神经网络的相互作用,提出了一种模块化免疫神经网络,该网络是一个动态平衡的、自适应的、分布式的网络模型。文献[93]结合免疫识别原理与神经网络特征,构造了一种免疫神经网络故障检测系统,并将其应用于捕获被检测队形的异常模式特征。文献[94]提出了基于免疫克隆聚类的协同神经网络原型向量求解算法,该算法充分利用免疫克隆的高效全局最优搜索能力来构造数据聚类,将构造的新聚类算法用于协同神经网络的原形向量训练求解。

1.2.2 混合免疫智能算法应用研究进展

1. 数值优化

现实优化问题大多可以归结为数值优化问题,数值优化问题可以描述为满足所有约束条件的解空间中寻找问题解最优的目标函数。随着人工免疫系统的发展,人们开始将免疫优化方法及其混合算法应用于数值优化问题求解,并取得了大量的研究成果。其中最具有代表性研究成果:戚玉涛等人在文献[12]中提出了一种基于塔式主从模式的分布式人工免疫系统模型,并基于此模型设计了一种分布式免疫记忆克隆选择算法应用于数值优化问题求解,并运用 Markov 数学原理证明了算法的全局收敛性。文献[13]提出一种针对多峰函数优化问题的免疫粒子群网络算法,该算法函数的全局最优及局部最优解的寻优效率均得到了提高。文献[14]针对函数优化问题提出了一种基于蚁群信息素的无导师学习机制,构造了一种基于信息素模因的免疫克隆选择算法,算法预先无须提供确定候选的局部搜索策略,实验结果表明,基于信息素模因的免疫学习机制借助信息素浓度的最优信息收敛到目标值,有效提高了免疫克隆选择算法对数值函数优化效率。文献[15]将正交试验设计方法引入免疫克隆选择操作中,提出了基于正交试验的全局搜索克隆选择算法。将该算法应用于经典函数优化中,实验结果表明该算法能够有效地保持算法种群的多样性,算法在搜索前期具有非常快的收敛速度,同时可以避免不成熟收敛。

2. 组合优化

组合优化问题，即在给定的约束条件下求出使目标函数极小（或极大）的最优变量组合，一般求解方法时间复杂度非常大，且随着问题规模的增大，其呈现出爆炸式增长。所研究的问题涉及社会经济、信息科学、工程技术、交通运输、通信网络、电子电路设计等诸多领域。近年来混合免疫智能算法为组合优化问题的高效求解提供了新的思路。如 Jiao 等人在文献[4]中提出了一种基于免疫遗传算法的 TSP 问题求解方法，提高了 TSP 问题的求解精度。为了提高大规模 TSP 问题的求解效率，文献[16]提出了塔式主从模型的两层并行人工免疫系统算法，并设计了分布式的免疫响应和免疫记忆机制。该算法运用疫苗的迁移来代替抗体的迁移，兼顾了种群多样性的保持和算法的收敛速度，最后将算法应用于大规模 TSP 组合优化问题求解，提高了大规模组合优化问题求解性能。文献[7]将免疫系统中受体编辑、疫苗接种等机制融入粒子群中，提出了基于免疫原理粒子群混合优化算法，并将该算法用于作业调度组合优化问题。Anna_Sswiecicka 等人在文献[17]中将自动机模型与人工免疫系统进行结合，提出了一种多处理器调度的混合免疫优化算法，提高了处理器执行任务的效率。

3. 多目标优化

多目标优化问题包含 n 个参数（决策变量）的集合，由一个包含 p 个目标的集合和一个含有 m 个约束条件的集合组成，其中目标函数和约束条件是决策变量的函数。近年来混合免疫智能优化方法在多目标优化问题得到了广泛的应用。如 Ashish Ahuja 等人在文献[5]中将免疫系统与蚁群算法进行融合，利用免疫算法优化信息素，提出了基于免疫蚁群混合优化算法的多目标问题优化方法。文献[18]结合量子信息处理与人工免疫系统，提出了量子免疫克隆选择算法，通过量子变异与重组来加快抗体成熟，并将该算法用于多目标问题求解，提高了多目标问题的全局求解能力。Felipe Campel 等人提出了一种基于多值免疫网络的多目标问题求解方法^[19]。Aldo Canova 在文献[20]中将混合免疫算法用于多目标天线布线组合优化，提高了天线布线求解效率。

4. 故障诊断

在实际工程应用中,各种工具或设备运行一段时间后,通常就会出现故障,为防止某些故障的扩大而带来的损失,有必要对系统进行故障诊断。人工免疫系统具有自动免疫识别能力,能为故障诊断所借鉴。近年来人们不断地从人工免疫系统中提炼出故障诊断方法,如 Anil Pahwa 等人在文献[1]中将小波分析与免疫神经网络结合用于分布式系统故障检测。Hao 等人在文献[21]中将免疫网络与分类算法结合研究提出了基于免疫网络分类算法的故障诊断方法并应用于电力转换器故障诊断,提高了对设备的故障诊断率。Slavisa 等人利用免疫系统二次响应能力及对处于异常的站点做出异常判断的自我免疫识别特征,提出了一种基于免疫系统的移动 Adhoc 网络异常行为故障检测系统^[22]。Rogerio 等人在文献[23]中提出了基于免疫系统的自动取款机自适应故障诊断系统,对自动取款机异常情况进行实时监控,提高了机器运行效率。

5. 计算机系统安全

随着计算机系统及互联网的高速发展,计算机系统安全日益成为突出的问题,而异常入侵检测、病毒防范等都可以从生物免疫机制中获得不少启发。文献[10]将免疫系统中的阴性选择原理用于计算机网络入侵检测,利用遗传算法对系统相关参数进行优化选择,提高了计算机系统入侵检测效率。Forrest 根据免疫系统阴性选择原理研究了一种用于检测异常数据变化的否定选择算法^[24],并将阴性选择机制用于解决计算机安全领域的问题。Paul K. Harmer 在文献[25]中依据生物免疫原理设计了一种分层分布式自适应多 Agent 计算机安全防御系统,其核心思想是利用免疫系统中自我识别与非我识别的否定选择原理,并将该模型应用于计算机安全监控,提高计算机网络病毒检测能力。Luther 在文献[26]中提出了一种合作型免疫入侵检测系统,提高了系统异常数据检测率。

6. 系统辨识

系统辨识是现代控制理论与应用研究的一个重要分支,也是控制理论能否成功应用于实际问题的关键技术之一,无论是在工程应用、物理学、生物学,还是在社会经济科学都有着广泛的应用。系统辨识过程是指在基于输入与输出观测的基础上,在指定的一类系统中,确定一个与被识别系统相等价的系统,并获得该系统的数学模型。近年来人们开始将混合免疫智能算法应用于系统参数辨识中,取得相关研究成果。例如,文献[27]借鉴免疫识别原理并结合递归模糊神经网络,提出了一种基于免疫递归模糊神经网络的复杂系统辨识方案。该辨识方案核心思想是将复杂系统模型分解为可变与不变两部分,并利用免疫遗传算法在线筛选出合适的构件来构造系统可变模型,以识别系统扰动。实验结果表明该方法能有效完成对复杂不确定系统的快速在线辨识。文献[28]提出了一种基于免疫粒子群优化算法的动态递归神经网络辨识器,并将其应用于超声马达非线性系统辨识中,实验结果表明:该方法所设计的辨识器和控制器在辨识与控制过程中不仅具有很高的收敛精度和速度,而且还对随机扰动信号具有较强的鲁棒性。文献[29]提出了一种基于混沌免疫粒子群算法与 El-man 神经网络耦合算法的非线性动态模型参数辨识方法,实现了辨识模型的输出逼近输入目标函数最优值,达到了非线性系统参数的准确辨识。文献[30]将免疫原理引入 RBF 神经网络中,提出了一种基于免疫原理的 RBF 神经网络辨识器,促使网络辨识器具有在线学习和记忆新样本的功能,并将该辨识器应用于某 300MW 火电机组主气压的多工况辨识中,实验结果表明该算法能够很好地适应对象的时变特性。

7. 智能控制

实际的工业控制对象往往具有非线性、不确定性、时变性等复杂特性,传统的控制方法难以获得更好的控制性能。将免疫系统机理引入控制领域,对解决复杂系统智能控制提供了新的思路。Cheng-Hung 等人提出了一种基于免疫聚类克隆选择算法优化设计的神经模糊控制器^[31],将该控制器应用于车辆控制,并取得了较好的控制效果。Amanda M 提出了基于增强学习的独特性免疫网络机器人控制方法^[32],通过分布性免疫网络对机器人进行协同控

制。Henry 在文献[33]中提出了一种免疫分布式多 Agent 协同控制框架,各个 Agent 相互通信协作,以应对动态环境进行协同控制。文献[34]提出了基于免疫粒子群优化的模糊神经网络控制方法并用于火炉温度系统解耦控制。文献[35]提出了基于免疫遗传算法的统一潮流控制器对多机电力系统进行控制,实验仿真结果表明,基于免疫遗传算法设计的统一潮流控制器能有效控制阻尼系统的低频振荡,提高电力系统的稳定性。

8. 复杂系统优化与控制

电力系统优化控制是典型的复杂系统优化控制问题,其研究的是电力系统最优运行状态,其模型中存在非线性、多变量、多约束、多目标的优化问题。近年来混合免疫智能优化方法开始应用于这一领域,为此类问题的研究提供崭新的技术路线。Liao 在文献[36]中将模糊决策与免疫退火选择方法融入免疫系统提出了混合免疫智能优化算法,并将该算法应用于电力系统机组启停问题,获得了机组的合理运行方案,在满足系统总负荷、保证供电质量的前提下系统燃料总耗量达到最优,运行安全,达到了系统最优运行要求。李玉龙等人在文献[37]中提出了基于相似性矢量距选择策略的免疫遗传算法,通过调节部分参数来调节算法在相似性与浓度之间的平衡度,整个算法继承了免疫算法与遗传算法各自的优点,将提出的免疫遗传算法用于解电力无功优化问题,并对 IEEE30 节点系统进行了仿真计算,实验结构表明,该方法取得了令人满意的结果。

9. 数据挖掘

在人工智能的领域研究中,知识的自动获取是最重要的关键技术,而数据挖掘是解决这一关键技术的主要方案。数据挖掘即从海量的数据中抽取隐含的、深层次的有用信息。近年来一些相关研究人员开始将混合免疫算法应用于数据挖掘中,目前基于免疫系统的数据挖掘方式主要为聚类、关联规则挖掘与数据分类。Alex A 在文献[38]中结合机器学习与人工免疫系统提出了一种面向问题的免疫系统数据挖掘思想,应用于数据聚类与分类及信息的提呈。Tien 等人提出了基于混合免疫系统的复杂数据集关联规则挖掘算法^[39],

该算法在时间性能与精确度上均优于以往关联规则求解算法。Muhammad 等人设计了一种基于免疫系统的模糊神经网络识别系统^[2]，对混合物气味进行分类识别。Seralozs 等人将经典核函数融入免疫系统^[40]，提出了一种基于核函数免疫系统的非线性系统分类方法。文献[41]提出了一种基因编码抗体群的免疫克隆选择算法，并将该算法应用于数据规则分类。

10. 图像处理

现代图像处理过程中涉及的内容主要包括图像分割、图像增强、图像恢复、图像压缩、图像重构以及图像融合等多方面，图像处理需要优化的参数较多，可以归结为复杂的优化问题，有的甚至是 NP-hard 问题。现代免疫优化方法的发展为解决这一难题提供了有力的工具。一方面人们利用混合免疫算法对图像进行分类处理，如，Zhong 等人提出一种免疫多值网络分类算法^[42]，利用网络分类算法对遥感图像数据进行分类处理，提高了对遥感图像的识别率。马文萍等人在文献[43]中将量子交叉操作引入人工免疫系统中，构造了一种基于量子克隆优化的 SAR 图像分类处理算法。文献[44]研究了一种基于模糊免疫网络的火焰数字图像分类研究方法。该方法充分利用模糊免疫网络算法不需要事先了解类别的先验知识，能够实时跟踪火焰燃烧状况的变化而实现动态聚类，克服了传统模糊聚类算法须预先确定聚类数的缺陷。实验结果证明了该方法能有效地判断火焰燃烧状态的改变。另一方面研究人员利用免疫算法对图像进行聚类处理，例如，文献[45]提出了一种基于混沌免疫算法的模糊聚类方法，该算法采用了模糊聚类的边缘检测策略，并结合人工免疫进化算法和混沌搜索算法各自的优点，有效实现了图像的边缘检测。

1.3 现有混合免疫智能算法存在的主要问题及解决途径

现有人工免疫系统与其他智能算法的结合研究与应用取得了一些成果，但还存在一些问题，主要表现为混合免疫智能算法理论研究还局限于低层次的简单混合，其应用研究还不够广泛。

1.3.1 混合免疫智能算法理论研究的缺陷及解决思路

现有混合免疫智能算法研究主要是将免疫原理引入其他算法中,主要是利用免疫系统中的抗体多样性机理去改善其他算法,促使其他算法跳出局部极值点。混合免疫计算智能的研究还停留在两种或者多种单一智能算法在一定程度上的简单“堆积”,并未完全考虑多种智能方法之间的有机联系及算法内在的差异性。混合免疫智能算法理论有待向深层次发展,需要深入挖掘免疫系统中的主要智能处理特征以及将免疫系统机理引入其他智能计算方法;或者将其他方法的主要特点和思想引入人工免疫系统中,优势互补,从而实现人工免疫系统和其他智能处理方法的真正融合;充分汲取各算法独特的优势,构造更高效的混合免疫算法。为了进一步丰富与发展混合免疫智能理论与算法,本书针对混合免疫智能算法的研究提出了四类解决思路。

第一,引入协同进化思想。混合免疫智能算法,通常以达尔文进化论作为生物理论依据,只考虑到个体适应度进化,而没有考虑到环境资源、群体之间竞争协作对种群进化的影响。生物学证据表明协同进化机理可以加快生物进化的历程,这一点在以往的混合免疫智能算法中很少得到体现。针对这一情形,引入生态学中的协同进化 Lotka-Volterra 思想到人工免疫算法中,考虑了个体与群体间、个体与个体、群体与群体之间的竞争合作关系,设计了局部免疫优势算子,构造了一种竞争合作型协同进化免疫克隆选择算法模型,以改善抗体的空间分布性,扩大解的搜索范围。

第二,不同的优化机理互补。人工免疫算法具有多样性能协助其他算法跳出局部极值,其他优化算法的优势机理或算子反过来也可以加速人工免疫算法的收敛性。利用系统的克隆选择和受体编辑机制能有效抑制算法收敛过程中的早熟停滞现象的优点,同时考虑粒子群的简单快速收敛性可以增强人工免疫算法的收敛效率,将粒子群体划分为两种动态模式,构造一种免疫双态粒子群优化算法模型,加快算法收敛速度,避免算法冗余迭代。

第三,不同优化机理集成融合。通过将免疫优化机理与其他不同优化机制进行集成融合,可以构成一种全局混合优化方法。融合免疫系统中克隆选择原理、免疫网络原理与协同进化思想及粒子群的邻域信息优点,构建了一种免疫协同粒子群进化算法模型,增强算法的全局和局部搜索效率,提高了

算法动态寻优性能。

第四,具有明显差异的优化行为互补。通过将免疫优化算法与不同优化行为的算法进行自适应融合,相互弥补自己的不足,构成一种强于原有算法性能新的混合优化算法。蚁群算法具有良好的反馈机制,运行前期收敛较快,但后期进化趋于停滞。人工免疫算法具有全局搜索能力,但对系统信息反馈利用不足,迭代寻优效率低。利用克隆选择算法与蚁群算法各自的优势,形成优势互补,构造了两类免疫克隆选择与蚁群自适应融合优化模型。

1.3.2 混合免疫智能算法应用研究的局限及拓展

混合免疫智能处理技术已经应用到许多问题领域,但是需要做更多的努力去理解混合免疫智能系统的本质,找到最适合的应用。从目前的研究状况看,建立统一的混合免疫智能算法理论体系难度还是比较大的,因此可以考虑从实际应用问题入手,开展更广泛的应用技术研究,如混沌系统的自抗扰优化控制、永磁同步电机系统多参数辨识与状态监测等复杂系统,混合免疫智能处理技术也具有很好的应用潜力。为了进一步充实和发展混合免疫智能算法的应用研究范畴,书中将所提混合免疫智能算法在两类新的领域中展开应用研究,试图突破现有混合免疫智能算法的应用局限性,促进其实用化。

第一,利用所研究的免疫双态粒子群优化算法具有强全局搜索能力及快速实时性优点,进行复杂系统控制器优化设计。其一,将免疫双态粒子群优化方法应用于混沌系统自抗扰优化控制中,构造一种混沌系统自抗扰优化控制器;其二,利用自抗扰控制器(ADRC)与小脑神经网络(CMAC)各自的优势构造 ADRC-CMAC 并行控制器,将免疫双态粒子群优化方法对并行控制器进行优化,构造出一种自抗扰神经网络并行优化控制方法。

第二,利用研究的免疫协同粒子群进化算法具有大解空间快速并行搜索能力,进行复杂非线性动态系统建模与参数辨识。将免疫协同粒子群优化方法应用于永磁同步电机系统多参数辨识中,构造一种基于免疫协同粒子群进化算法的永磁同步电机多参数辨识与状态监测模型方法。

所研究的混合免疫智能算法在混沌系统自抗扰优化控制及永磁同步电机系统多参数辨识这两类复杂系统问题中的成功应用拓宽了混合免疫智能算法的实际应用范畴;同时,混合免疫智能算法为这两类系统的深入研究提

供了新的思路。下面就混沌系统控制、永磁同步电机系统参数辨识这两类应用问题进行概述。

1. 混沌系统控制

混沌是非线性动力学所特有的一种运动形式,对初始值极为敏感,其行为轨迹非常复杂,同时混沌系统在数据加密^[95~97]、保密通信^[98~100]、混沌电路设计^[101,102]、生命医学^[103]等诸多领域有着广泛的应用。过去人们一直认为混沌是不可控制的,但是自从 Ott 等人提出了一种基于参数扰动的方法^[104],成功地实现了混沌系统的控制后,混沌系统的控制问题研究逐渐进入人们的视野,引起了研究人员浓厚的兴趣,许多研究人员进行了一些有意义的探索,新的研究成果不断出现。现有的控制方法主要有神经网络控制^[105]、自适应控制^[106, 107]、模糊控制^[108, 109]、智能 PID 控制^[110]等方法,这些方法提高了混沌系统的控制效果,但这些方法面对实际混沌系统时,待解决问题还有许多,且大多数控制方法需要混沌系统具有精确数学模型。而实际混沌系统的参数往往是部分未知或全部未知的,系统内部参数的摄动及外部的扰动也会给混沌系统带来不确定性,其系统数学模型难以精确,易导致控制性能失效^[111]。自抗扰控制器(ADRC)继承了 PID 控制器的优点,是一种不依赖于精确模型的非线性鲁棒控制器,具有控制精度高、响应速度快、抗干扰能力强等特点^[112],现已广泛地应用于控制工程中^[113],将自抗扰控制技术应用于混沌系统,构建混沌系统自抗扰控制器,将混沌系统变为规则运动是可行的。

2. 永磁同步电机系统参数辨识

永磁同步电机(PMSM)具有响应速度快、控制性能好、高密度功率等优点,业已广泛地应用于风力发电、电动汽车、精密伺服系统、船用推进系统等工业驱动系统中^[114]。研究表明 PMSM 系统参数易受温度、定子电流、磁通饱和等变化因素影响,造成磁链观测不准确和 d-q 轴电流无法彻底解耦,给控制系统带来了很大难度,降低了系统运行的可靠性^[115]。同时由于基于永磁同步电机的控制系统设计往往需要精确的电机参数值来辅助设计,例如,矢量控制调节器最优 PI 参数设计、无速度传感器的设计、定转子温度监控以及驱动系统故障诊断等。因此 PMSM 电机参数能否准确辨识影响到电机控制

性能以及电机状态的准确监测。现有永磁同步电机参数辨识方法主要如下。

① 固定永磁同步电机系统部分参数值为理论设计值,以减少待辨识参数个数。在文献[116]中,将辨识算法中永磁同步电机状态方程的部分参数值设置为固定的设计值来构建一个可辨识模型,再利用辨识算法直接估算方程中其余参数。此类方法实现简单,其缺点在于参数设计值并非就是实际值,所辨识的结果易偏离实际值。

② 让电机系统工作于不同的特殊工作点来减少待辨识参数个数,从而在不同的工作点求解不同的参数值。以文献[117]、[118]为例,其设定永磁同步电机处于停顿状态(0转速),从而不予考虑转子永磁磁链幅值的大小,通过注入扰动信号来辨识永磁同步电机的定子绕组电阻值和电感值。如文献[119]、[120]提出使用分步法来分别求解永磁同步电机的不同参数值,通过在一个状态下所辨识的参数值回代系统方程中再求解其余参数值。此类方案需要让永磁同步电机系统远离其工作点,因此应用范围具有一定的局限性。

③ 通过增加参数检测设备来减少永磁同步电机系统待辨识参数个数,从而构建一个满秩的参考/可变模型系统。工程领域常通过增加温度传感器、功率测量装置、转矩测量仪^[121~122]等获得更多额外电气信息来辅助辨识算法估算电机的参数值。该类方法须添加额外的精密测量设备,在求解过程中常需要综合考虑实际执行器件非线性因素补偿对参数辨识精度的影响^[123],常应用于一些大型应用领域。

④ 通过在永磁同步电机系统工作点附近注入扰动信号,获取更多有效系统状态方程来设计满秩系统^[124~126]。由于传统电机方程的阶数是二,而待辨识的参数有四个,相对于需要辨识的参数的个数来说,电机方程的阶数不够而非满秩。通过注入不同的扰动信号获得辨识满秩方程。此类方法不需要添加任何硬件,也不需要系统远离其工作点,因此该方法逐渐成为参数辨识研究的主流。

同时由于实际非线性永磁同步电机动态系统的强非线性、不稳定性、模型误差、测量误差、计算误差等各种“扰动”的存在等,常规方法难以得到高精度参数估计值。该类系统多参数辨识与状态估计仍然是一大理论与技术难题。

1.4 背景与意义

现代工业系统变得越来越复杂,而复杂系统的建模、优化与控制需要高性能的算法来辅助,依靠单一模式的优化方法难以满足系统性能要求。混合免疫智能处理技术为这类问题提供了有效的途径,同时也是人工免疫系统研究的发展方向。现有混合免疫智能算法研究主要是将免疫原理引入其他算法中,利用免疫系统中的抗体多样性机理去改善其他算法,促使其他算法跳出局部极值点,还停留在两种或者多种单一智能优化算法在一定程度上的简单“堆积”,并未完全考虑多种智能方法之间的有机联系及其算法内在的差异性。混合免疫计算智能有待向深层次发展,通过不断地汲取经典最优搜索方法及现代启发式算法的优点,各取所长,建立新模式的混合免疫智能计算方法已是人工免疫系统研究的发展方向。研究性能的混合免疫智能优化算法进一步丰富与发展混合免疫计算智能理论与方法。

混沌是非线性动力学所特有的一种运动形式,对初始值极为敏感,其行为轨迹非常复杂,同时混沌系统在数据加密、保密通信、混沌电路设计、生命医学等诸多领域有着广泛的应用。现有大多数控制方法需要混沌系统具有精确数学模型。而实际混沌系统的参数往往是部分未知或全部未知的、系统内部参数的摄动及外部的扰动也给混沌系统带来了不确定性,其系统数学模型难以精确,易导致控制性能失效。自抗扰控制器(ADRC)继承了PID控制器的优点,是一种不依赖于精确模型的非线性鲁棒控制器,具有控制精度高、响应速度快、抗干扰能力强等特点,现已广泛地应用于控制工程中,将自抗扰控制技术应用于混沌系统,构建混沌系统自抗扰控制器,利用混合免疫智能算法对自抗扰控制器进行优化设计,构建基于混合智能算法的混沌系统自抗扰优化控制方法具有重要的意义。

永磁同步电机(PMSM)具有响应速度快、控制性能好、高密度功率等优点,业已广泛地应用于风力发电、电动汽车、精密伺服系统、船用推进系统等工业驱动系统中。研究表明PMSM系统参数易受温度、定子电流、磁通饱和等变化因素影响,造成磁链观测不准确和d-q轴电流无法彻底解耦,给控制系统带来了很大难度,降低了系统运行的可靠性。同时由于基于永磁同

步电机的控制系统设计往往需要精确的电机参数值来辅助设计,例如,矢量控制调节器最优 PI 参数设计、无速度传感器的设计、定转子温度监控以及驱动系统故障诊断等。因此 PMSM 电机参数能否准确辨识影响到电机控制性能以及电机状态的准确监测。建立一种永磁同步电机智能辨识与状态监测模型方法具有重要的理论与实际意义。

人工免疫系统是一种具有高度分布性的复杂自适应学习系统,蕴藏着强大的信息处理能力与知识学习能力。免疫信息处理技术可以为复杂工业问题的求解提供解决方法。基于免疫系统的机理,深入挖掘免疫系统中蕴含的智能学习机制并结合现代智能优化控制方法的优势研究高效能免疫计算智能理论与算法及相关工业应用,进一步丰富发展计算智能理论与应用,对提高对复杂工业系统建模、优化控制的性能具有重要的理论与实际价值。

1.5 内容安排

全书首先从理论上研究了四类混合免疫智能算法,并对算法进行了测试、实验分析比较,证实了所提算法的有效性。接着,将所研究的混合免疫智能算法应用于混沌系统自抗扰优化控制和永磁同步电机系统多参数辨识与状态监测这两类典型复杂系统中,展开实际应用研究。全书共 7 章,具体内容安排如下。

第 1 章:从生物免疫系统的起源阐述了人工免疫的发展,陈述了人工免疫系统中典型的免疫智能算法,接着综述了现有的免疫混合算法及其应用研究现状,分析了现有混合免疫智能算法研究中存在主要问题并提出了解决思路;最后阐述了混沌系统控制与永磁同步电机系统多参数辨识与状态监测研究概况。

第 2 章:将自然界协同进化思想引入免疫系统中,研究了竞争合作型协同进化免疫克隆选择算法。各子种群内部采用局部最优免疫优势、克隆扩增、免疫变异等相关算子操作。引入信息熵理论来改善种群多样性,对高层记忆库进行免疫杂交提升操作,通过迁移操作实现了整个种群信息共享与协同进化。实验表明所提的协同免疫进化方法是有效的。

第 3 章:融合了人工免疫的多样性机理及粒子群的信息交互优点。通过将粒子群体分为捕食与探索两种状态,研究了一种免疫双模态粒子群优化方法。对处于不同状态的微粒采用不同搜索策略,同时对粒子群个体极值进行

克隆优化,对不活跃个体进行受体编辑。实验表明该算法收敛速度快,求解精度高,对高维及多模态优化问题的求解效果特别明显。

第4章:融合免疫系统优化原理、协同进化思想及粒子群的邻域信息的优势,构建免疫协同粒子群进化模型。普通微粒种群内部通过三种混合策略共同产生新个体,微粒个体极值采用小波学习以加快收敛速度,免疫克隆选择算法对记忆库进行精细搜索,扩大了算法解空间的搜索范围,提高了对复杂问题的优化能力,实验测试表明该算法收敛精度高且全局搜索能力强。

第5章:融合免疫克隆选择算法与蚁群算法的优势,研究了两类免疫蚁群自适应融合算法。第一类构建了一类基于统计学习的自适应免疫蚁群融合算法,通过混沌扰动、克隆扩增、免疫基因等相关算子的操作增强了克隆选择算法寻优效率;构造了局部最优搜索算子,设计了统计学习控制参数,实现了克隆选择与蚁群优化的有机结合。第二类研究了一种双态免疫优势蚁群算法,该算法将蚂蚁分成两种状态模拟自然界群智能分工协作职能,扩大了解的搜索空间,将禁忌表中的抗体通过克隆扩增、高频变异等免疫算子操作得到精英蚂蚁,再对抗体记忆库引入局部最优免疫策略,有效抑制了收敛过程中的早熟停滞现象。实验仿真表明两类免疫蚁群算法具有可靠的全局收敛性及较快的收敛速度。

第6章:所研究的混合免疫智能算法具有强全局搜索能力、快速实时性等优点,适合于复杂系统控制器优化设计。其一,将研究的混合免疫智能算法应用于混沌系统自抗扰优化控制中,对自抗扰控制器参数进行优化,应用于混沌系统控制,构建一种基于混合免疫智能算法的混沌系统自抗扰优化控制器;其二,利用ADRC与CMAC各自的优势并构造ADRC-CMAC并行控制器,利用混合免疫智能算法对ADRC-CMAC控制器参数进行自学习寻优,构造出一种自抗扰神经网络并行优化控制方法。实验验证了所提的混沌系统优化控制方法是有效的。

第7章:所研究的混合免疫智能算法具有大解空间快速并行搜索能力,适合于复杂非线性动态系统建模与参数辨识。构造了一种混合免疫智能算法的永磁同步电机多参数辨识模型方法。永磁同步电机参数辨识结果表明该方法不需要知道电机设计参数先验知识,能够有效地辨识电机电阻, $d-q$ 轴电感与转子磁链。同时,当电机参数发生变化时,依然能够有效地追踪该参数变化。在辨识出电机定子绕组阻值后,根据金属阻值与温度之间的线性原理间接计算定转子温度,从而实现永磁同步电机系统温度在线监测。

第2章

竞争合作型协同免疫进化算法

协同进化是借鉴生物学模型发展起来的一种基于多群体的进化方法，而人工免疫算法采用基于个体自身亲和度的进化模式没有考虑其进化的环境和个体之间的复杂联系。协同进化的引入可以克服现有人工免疫算法的缺陷。本章将协同进化与人工免疫系统进行结合研究，综合免疫算法与协同进化算法的优势，构造一种竞争合作型协同免疫进化算法模型，设计相关算子，最后对算法进行了理论分析及实验对比研究。

2.1 引言

人工免疫系统（Artificial Immune System, AIS）是模拟生物免疫系统机理而构造的高性能、自组织、鲁棒性好的人工智能系统。免疫克隆选择算法是人工免疫系统的核心算法之一，受生物免疫克隆选择学说的启发，De Castro 等人最先提出了一种较为简洁的免疫克隆选择算法（Clone Selection Algorithm, CSA），并成功地用于解决模式识别、数值优化和组合优化问题等^[53]，并建立了基于免疫克隆选择机理的搜索算法一般框架。至今人工免疫系统已应用到诸多领域，如组合优化问题^[20]、预测控制与系统辨识^[127]、数据挖掘^[38]、机器学习^[32]等问题。以往人工免疫算法基于个体自身亲和度的进化模式，而这种单个体进化模式并没有考虑个体进化与进化环境之间的复杂

关联。正因如此,它在应用中易表现出早熟收敛,且收敛速度较慢等缺陷,因此基于纯粹个体亲和度控制的抗体群进化机制难以获得最佳寻优效果。

协同进化是借鉴自然界生态学模型发展起来的一类新型的进化计算,最早的研究始于 Potter 和 Jong 的工作^[128],他们将协同进化思想引入遗传算法用于多目标优化问题求解,并取得了较好的效果,其核心思想为引入多个种群,多个种群通过适应度的关联共同求解。受多种群思想启发,多群体思想开始被引入人工免疫系统中,例如,文献[129]将多种群思想引入免疫算法,提出了基于多群体的自适应免疫进化计算模式,在一定程度上提高了人工免疫算法的求解性能。文献[74]提出了一种求解 SAT 问题的多种群量子免疫克隆算法。然而以往的多种群免疫克隆选择算法没有考虑其进化环境与个体之间的复杂联系对个体进化的影响,以及群体之间的协同合作及资源竞争关联对算法的影响,这些问题是纯粹多种群免疫算法的一个缺陷。借鉴协同进化的思想和依据生物种群密度的机理而构建的竞争协同进化模型,能够很好地描述个体与资源环境、个体与群体以及群体与群体之间的协同竞争行为,基于此模型而设计的改进型免疫克隆选择算法在一定程度上能够突破只采用单个体适应度控制进化的现有生物进化框架局限。

借鉴生态学的种群密度^[130] Lotka-Volterra 协同进化思想及结合人工免疫系统的优势,本章提出协同免疫进化算法:竞争合作型协同进化免疫优势克隆选择算法模型(Competitive-Cooperative Coevolutionary Immunodominance Clone Selection Algorithm, CCCICA),算法由一个优势代表记忆种群与若干个普通种群构成。优势代表记忆种群充当抗体子种群领导集合,并对优势代表记忆种群进行免疫杂交提升操作。对每个普通子种群内部进化时,提出了一种改进的免疫克隆选择算法——局部最优免疫优势克隆选择算法(Immunodominance Clone Selection Algorithm, ICA),抗体群内部通过局部最优免疫优势、克隆扩增、动态高频混合变异等算子操作,计算基因座信息熵来判断和调控种群的多样性。通过迁移操作将优秀个体从记忆库中返回到低层各子种群中,成功模拟了自然界不同种群间的协作与竞争及信息交流现象,提高了算法的全局寻优能力。针对 TSP 多个测试实例实验结果表明:该算法在收敛速度与最优解等方面均取得了较好的效果。

2.2 竞争合作协同免疫进化模型

协同进化算法是在自然界中物种协同进化论基础上演变成的一类新的进化算法，其与一般进化算法的区别在于：协同进化算法在传统进化算法的基础上，考虑了种群与种群、种群与环境资源，其进化协调机制优于单种群进化。基于不同生物模型，协同进化算法可以分为以下几类^[131]：

- ① 基于种间竞争机制的协同进化算法；
- ② 基于“捕食-猎物”机制的协同进化算法；
- ③ 基于共生机制的协同进化算法。

同时协同进化算法中最常见的模型分为竞争型协同进化和合作型协同进化，以及竞争合作型协同进化模型^[132]。

借鉴自然界协同进化的思想和依据生物种群密度的概念而建立的竞争合作型协同进化模型，能够很好地理解个体与群体之间、个体与资源环境之间的竞争协同行为，基于此模型而设计的改进型免疫克隆选择算法不仅在一定程度上能够克服基于单个体适应度控制的生物进化框架模式的缺陷，而且也能进一步理解生物免疫系统细胞器官之间的相互协同过程。借鉴协同进化思想和生态学的种群密度^[130]思想，本章提出了一种竞争-合作型协同进化免疫算法，设计了竞争-合作协同进化免疫算法模型。该算法把总群体分割成若干子群体，每个子种群独立地进行免疫操作，底层子种群间采用基于生态种群密度的种群竞争协同模型，高层优势代表记忆库种群采用免疫杂交操作，通过迁移操作把全局最优值返回到各子种群中。促进了各种群间合作协同进化。本章设计的竞争协同进化模型，每个子种群都包含完整的解而不是解的分量，可避免在算法设计过程中使用过多的种群，以降低算法的复杂度。竞争合作免疫协同进化模型如图 2.1 所示。

由图 2.1 可知，竞争合作协同进化模型（CCCICA 模型）是由一个优势代表种群 S^* 和若干个抗体子种群 $S=(S_1, S_2, \dots, S_n)$ 组成，其中 S_i 代表第 i 个子种群，按照图 2.1 中的方式组织而成，其下层是 n 个规模为 N_i 的抗体子种群， N_i 采用基于生态种群密度的种群竞争调节机制（见 2.2.1 节）。优势代表记忆种群 $S^*=(S_{11}, S_{12}, \dots, S_{ij})$ 看成 $i \times j$ 个优势抗体。 S_{ij} 代表第 i 个子种群的第 j 个代

表。优势个体代表的提取：在每个子种群选择一个最好的个体（亲和度最高的个体）和随机选择一个次优个体。优势代表种群充当多种群的记忆库功能，承担了抗体种群间信息交互的任务。利用免疫遗传交叉操作对优良种群进行优化提升操作，种群中较有潜力的个体得到迅速优化，提升操作不仅保存了各子种群在进化过程中产生的好的模式，通过迁移操作将全局最优的个体返回到各子种群中进一步加速了它们的收敛速度。

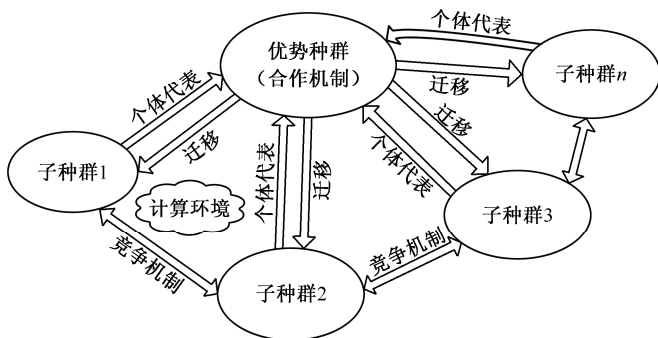


图 2.1 竞争-合作型免疫协同进化模型

2.2.1 基于生态种群密度的种群竞争操作

生态学理论认为生物个体在自身进化过程中不仅与个体适应度有关，而且还受所处生存环境、同其他个体之间的相互竞争等因素的影响。也就是说，在一定生态环境中的生物群落，其种群进化状态既受到自身适应度的影响，同时还受到生态资源环境以及自身群体与其他种群之间的所发生竞争协同作用等关联的影响，其中环境和种群间的协同竞争关系可以通过种群密度来体现。如果存在 n 个种群 $S_i (i=1, 2, \dots, n)$ ，则种群 S_i 的增长符合如下 Lotka-Volterra 竞争方程^[130]：

$$\frac{dN_i}{dt} = r_i N_i \left[\frac{K_i - N_i - (\sum_{j=1}^n a_{ij} N_j)}{K_i} \right] \quad (2.1)$$

式 (2.1) 中 N_i 表示种群 i 的大小， K_i 表示在不发生竞争的情况下种群 i

的环境负荷量, r_i 表示种群 i 的个体的最大瞬时增长率, a_{ij} 是竞争系数, 表示种群 j 的每个个体对种群 i 的抑制作用, $\frac{dN_i}{dt}$ 表示种群 i 的密度变化量。这是基于生态种群密度的种群竞争协同模型。把这种模型引入人工免疫算法中可以得到协同免疫进化算法, 该算法充分考虑了环境和种群间相互竞争的关系, 它能够提高个体的多样性, 增强算法的全局收敛性, 进一步理解生物免疫系统细胞器官之间的协同合作。子种群之间通过竞争不断调整各自抗体群的规模, 具体方法如下。

① 如果种群 $\frac{dN_i}{dt} > 0$, 通过随机产生 $\frac{dN_i}{dt}$ 个抗体加入种群 N_i 来增大该种群的规模, 有利于提高该子种群的多样性, 从而在一定程度上提高了抗体群的全局分布性。

② 如果种群 $\frac{dN_i}{dt} < 0$, 按照计算出来的适应度对种群 N_i 排序, 删除适应度最小并且相对于种内密度较大的 $\frac{dN_i}{dt}$ 个抗体, 从而减少该种群的规模, 一定程度上提高了该抗体种群的总体适应度。因此, 在下一轮的循环中, 可能显示出较强的竞争实力。

③ 依照式 (2.1) 调整各个子种群抗体群的规模: $N_i(t+1) = N_i(t) + \frac{dN_i}{dt}$ 。

这种竞争机制既体现了抗体子群进化过程中的种间协同竞争, 也体现了抗体子群内部抗体之间的相互竞争过程。

2.2.2 群体协同合作操作

优良记忆种群由各个普通种群中亲和度较优的若干个体组成, 是一个优良记忆库, 供各种群的进化进程共享。群体间协同合作主要包括:

- ① 优良抗体免疫杂交;
- ② 个体迁移操作。

1. 优良抗体免疫杂交

杂交策略是建立在子种群划分基础之上的一种新的优化策略，核心思想是在子种群间建立繁殖机制，模拟自然界同物种间不同种群间的杂交，完成种群间信息交互。借鉴遗传算法的杂交策略，还必须考虑抗体间的差异度。选择抗体时避免两个抗体过于相似而“近亲繁殖”，杂交过程必须保持两个抗体亲和力存在一定的差异。对记忆库优势抗体群 $S^*=(S_{11}, S_{12}, \dots, S_{ij})$ 进行免疫杂交，免疫杂交方式如下。

if $(|\text{affinity}(A_i) - \text{affinity}(B_i)|) > \varepsilon$ then crossover(). $\text{affinity}(A_i)$ 为抗体 A_i 的亲和力， ε 为很小的正常数免疫阈值。免疫杂交操作采取最优保存，杂交后生成的抗体将代替原始抗体，以保证子种群抗体数量稳定。为了确保优秀模式在子种群间传递，算法采用最优杂交策略，选取优势代表库中的待杂交抗体 S 与来自不同子群最优解抗体进行免疫杂交操作。

2. 优势抗体迁移

普通种群在自己的进化进程中，每隔一定的代数直接从优良种群中引进若干优良种子个体（一般取 1、2 个抗体）代替本种群中的较劣个体。迁移操作如下。

if $((\text{gen} \% \text{migration_interval}) == 0)$ then

Migrate(sig, Num_migrant); 其中 sig 为迁移信号，见式 (2.2)，当 sig=1 时表示迁移。

$$\text{sig} = \begin{cases} 1, \frac{F_{\text{best}}^i(k) - F_{\text{best}}^i(k-1)}{F_{\text{best}}^i(k-1) - F_{\text{best}}^i(k-2) + \varepsilon} < 1 \\ 0, \text{其他} \end{cases} \quad (2.2)$$

其中， $F_{\text{best}}^i(k)$ 表示第 i 个种群第 K 代亲和力最优的抗体， ε 为平滑系数。该式表明：种群 i 在进化的过程中连续出现了连续减慢现象，即需要从优良种群库中迁移优良抗体来提高该子种群的优化性能。

2.3 种群内部免疫优势克隆选择操作

定义 2.1: 免疫克隆操作。

克隆操作 $T_c^C, Y(t)=T_c^C(A(t))=[T_c^C(A_1(t))T_c^C(A_2(t)), \dots, T_c^C(A_n(t))]$, 其中 $Y_i(t)=T_c^C(A_i(t))=I_i \times A_i(t), i=1, 2, \dots, n$, I_i 为元素 1 的 q_i 维行向量, $q_i \in [1, B_i]$ 称抗体 A_i 的 q_i 克隆。克隆扩增后得到的新抗体种群为 $Y(t)=[Y_1(t)Y_2(t), \dots, Y_n(t)]$, 其中 $Y_i(t)=\{Y_{ij}(t)\}=[Y_{i1}(t)Y_{i2}(t), \dots, Y_{iq_i}(t)], Y_{ij}(t)=A_{ij}(t)=A_i(t), j=1, 2, \dots, q_i$ 。

$$B_i = \text{round}(\alpha \cdot \frac{\text{affinity}(A_i)}{i \cdot \sum_{i=1}^N \text{affinity}(A_i)} + b) \quad (2.3)$$

式中 α 为 ≥ 1 的比例系数, $\text{round}()$ 取整, b 为 >1 的整数常量。上式表明克隆操作后的抗体倍数与亲和力成正比, 与升序排序的位置成反比; 同时, 为了保证每个抗体都有一定数量克隆机会, 故加上了常量 b 。

定义 2.2: 免疫选择操作。

免疫选择 $T_s^C, P_i(t)=T_s^C(P(t))=\max\{\text{affinity}(Z_i(t))\}$, 然后 $P_i(t) \cup A_i(t) \rightarrow A_i(t+1)$, 这样通过局部择优实现了种群的压缩映射, 同时保证了抗体群中的最优解不会变差。

2.3.1 抗体局部最优免疫优势算子

现代免疫学研究表明^[8], 一个抗原分子上可以有多个表位, 但在诱导宿主免疫应答时只有一种或一个表位起主要作用, 使宿主产生以该特异性为主的免疫应答反应, 这种特性称为免疫优势现象。例如以 TSP (旅行商) 问题为例, 本章设计一种针对 TSP 问题的免疫优势算子, 以提高对该类问题的求解精度, 加快算法收敛速度。TSP 问题的抗体局部最优免疫优势算子 (LOIO) 构造原理如下。

设有 n 个节点 (城市号), $e_1=(v_1, v_2, \dots, v_i, v_{i+1}, \dots, v_n) \subset a_p, a_p \in Ab$ (抗体群), 设有待插入节点 $v_j \in e_2 \subset a_p, e_1, e_2$ 是抗体 a_p 的两个子集 (也称作用域), 且满足 $e_1 \cap e_2 = \emptyset$ 。ID = $\{(\Delta d_i, v_i, v_j, v_{i+1})\}$ 为抗体局部最优免疫优势

算子。根据计算几何学三角形最优距离计算，其 $(\Delta d_i, v_i, v_j, v_{i+1})$ 满足下式：

$$\Delta d_i = \min(d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1})) \quad (2.4)$$

其中 $v_i, v_{i+1} \in e_1, v_j \in e_2$ 。

定理 2.1: 设 n 点（城市号） $e_1 = (v_1, v_2, \dots, v_i, v_{i+1}, \dots, v_n) \subset a_p$, $a_p \in Ab$, 设有待插入节点 $v_j \in e_2 \subset a_p$, e_1, e_2 是抗体 a_p 的两个子集（也称作用域），满足 $e_1 \cap e_2 = \emptyset$ 。如果有 v_j 插入 v_i 和 v_{i+1} 之间，使得路径长度 $\text{dist}_i = d(v_1, v_2) + d(v_2, v_3) + \dots + d(v_i, v_j) + d(v_j, v_{i+1}) + \dots + d(v_{n-1}, v_n)$ 最短，那么就等价于 $\Delta d_i = d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1})$ 最短。

证明如下：

① 将 v_j 插入 $v_i (i \leq n)$ 之后有如下路径：

$$\bullet v_1 \cdots \bullet v_i \rightarrow \bullet v_j \rightarrow \bullet v_{i+1} \cdots \bullet v_k \rightarrow \bullet v_{k+1} \cdots \bullet v_n$$

得路径长度

$$\text{dist}_i = d(v_1, v_2) + \dots + d(v_i, v_j) + d(v_j, v_{i+1}) + \dots + d(v_k, v_{k+1}) + \dots + d(v_{n-1}, v_n)$$

② 若将 v_j 插入到任一节点 $v_k (k \leq n)$ 之后有如下路径：

$$\bullet v_1 \rightarrow \bullet v_2 \cdots \bullet v_i \rightarrow \bullet v_{i+1} \cdots \bullet v_k \rightarrow \bullet v_j \rightarrow \bullet v_{k+1} \cdots \bullet v_{n-1} \rightarrow \bullet v_n$$

得路径长度

$$\text{dist}_k = d(v_1, v_2) + \dots + d(v_i, v_{i+1}) + \dots + d(v_k, v_j) + d(v_j, v_{k+1}) + \dots + d(v_{n-1}, v_n)$$

③ 如果 $\text{dist}_i < \text{dist}_k$ ，即有 $\text{dist}_i = \text{dist}_k < 0$ ，则

$$\begin{aligned} & (d(v_1, v_2) + \dots + d(v_i, v_j) + d(v_j, v_{i+1}) + \dots + d(v_k, v_{k+1}) + \dots + d(v_{n-1}, v_n)) - \\ & (d(v_1, v_2) + \dots + d(v_i, v_{i+1}) + \dots + d(v_k, v_j) + d(v_j, v_{k+1}) + \dots + d(v_{n-1}, v_n)) < 0 \end{aligned}$$

$$\begin{aligned} \Delta d_i &= (d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1})) < \Delta d_k \\ &= (d(v_k, v_j) + d(v_j, v_{k+1}) - d(v_k, v_{k+1})) \end{aligned}$$

反之，如果有

$$\begin{aligned} \Delta d_i &= (d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1})) < \Delta d_k \\ &= (d(v_k, v_j) + d(v_j, v_{k+1}) - d(v_k, v_{k+1})) \end{aligned}$$

则有 $\text{dist}_i < \text{dist}_k$ 。

④ 由上述可得

$$\Delta d_i = \min(d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1}))$$

其中 $v_i, v_{i+1} \in e_1, v_j \in e_2$, 对应的 v_i, v_j, v_{i+1} 就是抗体中的免疫优势位。该算法能保证待插入节点 v_j 插入的位置是 e_1 中最优的位置, 即使得路径长度

$$\text{dist}_i = d(v_1, v_2) + d(v_2, v_3) + \cdots + d(v_i, v_j) + d(v_j, v_{i+1}) + \cdots + d(v_{n-1}, v_n)$$

最短。这样只通过计算 v_i, v_j, v_{i+1} 之间的路径长度 Δd_i , 就可以避免计算复杂的 $e'_1 = (v_1, v_2, \cdots, v_i, v_j, v_{i+1}, \cdots, v_n)$ 的总路径长度 dist_i , 大大减少了计算量。局部最优免疫优势算子构造如下。

局部最优免疫优势算子 (LOIO) 伪代码:

begin:

{while($e_2 \neq \Phi$)do{

 求抗体免疫优势 $ID = \{(\Delta d_i, v_i, v_j, v_{i+1})\}$ 其中 $\Delta d_i = \min(d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1}))$;

$v_i, v_{i+1} \in e_1, v_j \in e_2$;

 将 v_j 插入 v_i 之后, $e_1 = e_1 \cup v_j$

 从 e_2 中移出 $v_j, e_2 = e_2 - v_j$

 }

}

end

具体地, 对于表 2.1 所示的 10 个城市 TSP 问题 (表中给出的是任意两个城市之间的距离), 抗体免疫优势算子的作用过程如下。

表 2.1 城市之间的距离信息

D_{ij}	1	2	3	4	5	6	7	8	9	10
1	0	7	6	4	12	9	7	1	12	8
2	7	0	2	5	18	15	14	7	18	14
3	6	2	0	5	17	14	13	6	17	13
4	4	5	5	0	14	12	11	4	15	12
5	12	18	17	14	0	5	6	11	9	9
6	9	15	14	12	5	0	2	8	5	5
7	7	14	13	11	6	2	0	8	5	3
8	1	7	6	4	11	8	8	0	12	9
9	12	18	17	15	9	5	5	12	0	4
10	8	14	13	12	9	5	3	9	4	0

Step 1: 挑选两个抗体（一个亲和度最高的抗体 t_2^{old} 和一个将做免疫优势的抗体 t_1^{old} ）中的片段作为免疫操作的区域：

$$\begin{aligned} t_1^{\text{old}} &= 8\ 5\ 6\ |\ 9\ 1\ 4\ 2\ |\ 10\ 3\ 7 \\ t_2^{\text{old}} &= 2\ 4\ 1\ |\ 5\ 7\ 10\ 6\ |\ 9\ 8\ 3 \end{aligned}$$

Step 2: 将 t_2^{old} 的作用区域加到 t_1^{old} 的前面， t_1^{old} 的作用区域加到 t_2^{old} 的前面，然后自作用区域后依次删除与作用区相同的城市码，得到如下抗体：

$$\begin{aligned} t_1^{\text{mid}} &= 5\ 7\ 10\ 6\ |\ 8\ 9\ 1\ 4\ 2\ 3 \\ t_2^{\text{mid}} &= 9\ 1\ 4\ 2\ |\ 5\ 7\ 10\ 6\ 8\ 3 \end{aligned}$$

Step 3: 遵循抗体局部最优免疫优势的定义，改变抗体的某些位，例如将城市 5 插到城市 8 后面，城市 9 插到城市 10 后面，依次类推，得到新的抗体：

$$\begin{aligned} t_1^{\text{new}} &= 8\ 5\ 6\ 9\ 10\ 7\ 1\ 4\ 2\ 3 \\ t_2^{\text{new}} &= 5\ 7\ 10\ 9\ 6\ 8\ 3\ 1\ 4\ 2 \end{aligned}$$

显然，免疫优势算子操作通过两个抗体的相互作用，并依据抗体免疫优势来确定基因的相对位置，以提高这两个抗体的亲和度，是对特定抗体的一个局部最优动态调整过程。它包括两部分：一是基本免疫优势的形成，二是抗体免疫优势的获得。前者是从具体问题中获得先验知识的过程，而后者则是先验知识运用的过程。通过免疫优势可以加速群体亲和度成熟，加快算法收敛，降低 CPU 实际时间代价消耗。在这里须说明的是即使两个相同的抗体通过免疫优势操作后可以得到两个不同的抗体，同样可以增强抗体群多样性。

2.3.2 基于信息熵的种群多样性控制

平均信息熵，假设抗体群由 N 组抗体组成，每组抗体有 L 位基因，设编码字符集大小为 $|K|$ ，其中 $K=\{k_1, k_2, \dots, k_{|S|}\}$ ，则对应于基因座 j 的信息熵定义为

$$H_j(N) = -\sum_{k=1}^{|K|} p_{kj} \log_2(p_{kj}) \quad (2.5)$$

上式表明 $H_j(N)$ 为第 j 个基因座的总体信息测度, P_{kj} 为第 k 个符号在基因座 j 上的概率 $P_{kj} = \frac{n_{kj}}{N}$; 其中 n_{kj} 为在基因座上出现第 k 个符号总个数。群体的平均信息熵定义为 $H = \frac{1}{L} \sum_{j=1}^L H_j(N)$ 。上述表明种群的平均信息熵越大, 种群的分布性越好。现有研究表明随机进化算法随着进化代数 t 的增加, 群体适应度值不断提高, 群体中个体状态逐渐趋同, 多样性逐渐降低, 当群体中个体趋于一致时, 多样性为零。

1. 多样性判断

每代的多样性阈值定义如下:

$$\phi(t) = A \cdot \log_2(|K|) \cdot e^{\frac{Bt}{G}} \quad (2.6)$$

式中, $A \in (0,1)$ 为调节系数, $B > 0$ 为加速因子, t 为当前进化代数, G 为总的进化代数。为了防止算法进化过程中种群趋同而陷入局部最优, 在算法迭代执行过程中加入多样性判断, 若种群多样性低于预先设定的阈值, 则需要对群体多样性进行调控以改善种群的多样性。

2. 多样性调控操作

在子种群中 $N_b(N_b > 1)$ 个亲和度相同抗体 (一般为最优的抗体, 或次优个体) 里选择 $N_b - 1$ 个抗体, 对所选择低多样性个体进行基因倒位操作, 直到这些抗体基因位的多样性超过预定阈值。

2.3.3 种群内部抗体克隆选择操作

种群内部抗体克隆选择操作如图 2.2 所示。

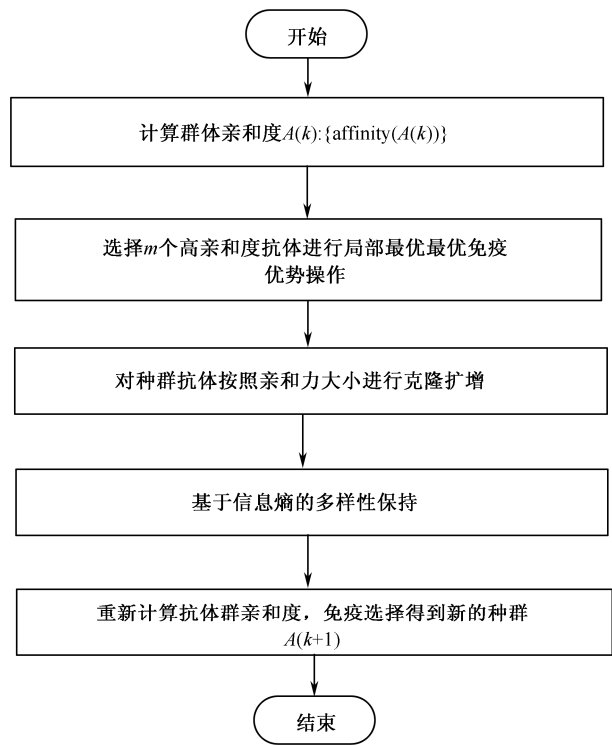


图 2.2 种群内部抗体免疫优势克隆选择操作

2.4 算法流程及其收敛性能分析

2.4.1 竞争合作型协同免疫进化算法步骤

竞争合作型协同免疫优势克隆选择算法的伪码描述如下。

```
begin
    步骤 1: 种群初始化及相关参数初始化
    步骤 2: for 对每个子种群  $S_i(i=1,2,\cdots,n)$ , do {
        步骤 3: 利用 Lotka-Volterra 竞争方程, 计算抗体群  $S_i(i=1,2,\cdots,n)$  的增长值  $\frac{dN_i}{dt}$ ;
```

① 如果种群 $\frac{dN_i}{dt} > 0$, 通过随机产生 $\frac{dN_i}{dt}$ 个抗体加入种群 N_i 来增大该种群的规模;

② 如果种群 $\frac{dN_i}{dt} < 0$, 按照计算出来的适应度大小排序, 删除适应度最小的 $\frac{dN_i}{dt}$ 个抗体, 从而减少该种群的规模 (为了防止种群规模等于 0, 种群规模 $N_i(t) > 0$) $N_i(t+1) = N_i(t) + \frac{dN_i}{dt}$;

步骤 4: 计算抗体群亲和度并对亲和度进行排序, 选择前 K 个高亲和度抗体进行局部最优免疫优势操作, $S'_i(t) = \text{LOIO}(S_i(t))$;

步骤 5: 抗体亲和度进行排序, 按照亲和度大小克隆扩增, $S''_i(t) = T_c^C(S'_i(t))$;

步骤 6: 执行免疫基因操作, $Z_i(t) = T_g^C(S''_i(t))$;

步骤 7: 计算亲和度进行免疫选择得到新的抗体群, $S_i(t+1) = T_s^C(Z_i(t)) = \max\{\text{affinity}(Z_i(t))\}$;

步骤 8: 抗体优势代表 S_{ij} 的提取, $S^*(t) = S^*(t) \cup S_{ij}$;

步骤 9: 优良记忆库免疫杂交提升操作;

步骤 10: 优势抗体迁移操作;

步骤 11: $t = t+1$ 。

end

2.4.2 收敛性分析

定义 2.3: 设 $S^N = \{x = (x_1, x_2, \dots, x_N), x_i \in S(i \leq N)\}$ 为抗体种群解空间, X_i 表示个体, S 表示个体空间。适应度值函数 $f: S \rightarrow R^+$, 为个体空间到正实数空间的映射, 则存在全局最优解集合为 $M = \{x; \forall y \in S, f(x) \geq f(y)\}$, 满意解集合 $B = \{x \in B; \forall y \notin B, f(x) \geq f(y)\}$ 。

引理 2.1: 文献[8]研究表明基于免疫克隆选择策略算法的种群序列 $\{A(n), n \geq 0\}$ 是有限齐次马尔可夫链, 并且以概率 1 收敛。

定理 2.2: 基于抗体免疫优势算子操作的免疫克隆选择算法的优化解序列 $\{x(n), n \geq 0\}$ 是有限齐次马尔可夫链。

证明: 由引理 2.1 可知, 基于免疫克隆选择算法的抗体种群序列 $\{A(n), n \geq 0\}$ 是有限齐次马尔可夫链, $A(n+1)$ 仅与 $A(n)$ 有关, 与算法迭代次数 n 无关; 即有 $x(n+1) = x_i(n+1) = T_s^i \circ T_m^i \circ T_c^i \circ T_g^i(x(n))$, T_g^i 、 T_m^i 、 T_c^i 、 T_s^i 分别表示抗体免疫优势操作、克隆扩增操作、克隆变异操作、克隆选择操作, \circ 表示运算合成符号。以上操作算子均在有限状态空间中进行, $x(n+1)$ 仅与 $x(n)$ 有关, 且与算法迭代次数 n 无关。因此, 基于抗体免疫优势操作的克隆选择算法的优

化解序列 $\{x(n); n \geq 0\}$ 是有限齐次马尔可夫链。

定理 2.3: 基于抗体免疫优势操作的克隆选择算法的优化解马尔可夫序列是以概率为 1 收敛到满意解集 B 的子集 B_0^* , 这里定义 $B_0^* = \{y = (y_1, \dots, y_N); y_N \in M\}$, 即有

$$\lim_{n \rightarrow \infty} p\left\{\frac{x(n) \in B_0^*}{x(0) = x_0}\right\} = 1 \quad (2.7)$$

证明: 设 x 是 $f(x)$ 的唯一最优解, 由定义 2.3 和定理 2.2 可知 $P\{x, y\}$ 有如下性质:

① 当 $x, y \in B_0^*$ 时, 有 $P\{x, y\} > 0$, $P\{y, x\} > 0$, 即 $x \leftrightarrow y$;

② 当 $x \in B_0^*$, $y \notin B_0^*$ 时, 有 $P\{y, x\} = 0$, 即 $x \rightarrow y$ 。

可以得到, B_0^* 为正常返的非周期不可约闭集, $S^M \setminus B_0^*$ 为非常返的状态集。

$$\lim_{n \rightarrow \infty} p\left\{\frac{x(n) = y}{x(0) = x_0}\right\} = \begin{cases} \Pi(y), & y \in B_0^* \\ 0, & y \notin B_0^* \end{cases}$$

故有

$$(\lim) p\left\{\frac{x(n) \in B_0^*}{x(0) = x_0}\right\} = 1$$

定理 2.4: CCCICA 以概率 1 收敛于全局最优解。

证明: 不妨设问题 P 为 CCCICA 算法待求解的问题, f^* 为其全局最优适应值, 即 $f^* = \max\{f(x) | x \in P\}$ 。

则可令 t 时刻总进化群体表示为 n 个子种群的集合 $\{S_1(t), S_2(t), \dots, S_n(t)\}$, 优秀抗体集合为 $\{f_1^*(t), f_2^*(t), \dots, f_s^*(t)\}$ 。

假设

$$Z(t) = \max\{f_1^*(t), f_2^*(t), \dots, f_s^*(t)\}$$

于是有

$$\text{待证命题以 } \lim_{t \rightarrow \infty} P(z(t) = f^*) = 1$$

从算法中优势代表记忆库种群的保留、更新规则以及优秀抗体迁移等操作可知, 若在进化过程中存在某普通子种群收敛于 f^* , 则肯定全局最优值收敛于 f^* , 即 CCCICA 全局收敛的概率大于或等于任意普通子种群收敛于全局最优解的概率。

即对于任意的子种群 P^* 有

$$\lim_{t \rightarrow \infty} P(z(t) = f^*) \geq \lim_{t \rightarrow \infty} P(\max\{f(x) | x \in P^*(t)\} = f^*)$$

CCCICA 中各普通子种群内部的抗体免疫优势算子的克隆选择算法以概率为 1 收敛到种群内最优值 (定理 2.4 已得到证明)。若只要存在某一种群 p , 其种内最优适应值等于全局最优适应值 f^* , 即 $f^* = \max\{f(x) | x \in P^*\}$, 而且 CCCICA 中的成熟种群即为趋于收敛的种群。因此, 若 P^* 在进化过程中成为成熟种群, 收敛到其种内最优解 f^* 时, 即有

$$\lim_{t \rightarrow \infty} P[\max\{f(x) | x \in P^*(t)\} = f^* / P^*] = 1$$

又对于子种群 P^* , 有

$$\lim_{t \rightarrow \infty} P(P^*) = 1$$

综上所述, 于是有

$$\begin{aligned} 1 &\geq \lim_{t \rightarrow \infty} P(z(t) = f^*) \geq \lim_{t \rightarrow \infty} P(\max\{f(x) | x \in P^*(t)\} = f^*) \\ &\geq \lim_{t \rightarrow \infty} P[\max\{f(x) | x \in P^*(t)\} = f^* / P^*] \times \lim_{t \rightarrow \infty} P(P^*) = 1 \times 1 = 1 \end{aligned}$$

即

$$\lim_{t \rightarrow \infty} P(z(t) = f^*) = 1$$

2.4.3 时间复杂度分析

抗体子种群免疫优势的时间复杂度 $T(n) = O(N_c \cdot m \cdot n^2 \log(n))$, m 为抗体个数, n 为基因长度即为城市个数, N_c 为循环次数, 免疫选择、变异等其他操作的最坏时间复杂度为 $O(N_c \cdot m \cdot n^2)$ (m 为抗体个数)。高层优势抗体代表免疫杂交时间复杂度为 $O(m_0 \cdot n^2)$ (m_0 为优势抗体个数)。文中算法的时间复杂度最坏为

$$T(n) = O(N_c \cdot m \cdot n^2) + O(N_c \cdot m \cdot n^2 \log(n)) + O(N_c \cdot m \cdot n^2) + O(m \cdot n^2)$$

2.5 实验仿真

TSP 问题是著名的 NP 难问题, 其求解方法在集成电路芯片设计、网络路由、车辆路径规划等领域有着广泛应用前景^[133], 解决它对于解决一类 NP 难问题具有重要的理论价值和实际意义。其求解难度大, 一般常规方法难以

获得满意解。该问题通常也作为各种非线性优化方法的评价标准。为了验证本章所提算法的有效性, 选用国际上通用的 TSPLIB 标准测试库中的多个实例进行测试。

TSP 问题可以简单描述为: 已知 N 个城市 $v=\{v_1, v_2, v_3, \dots, v_N\}$, 以及任意两城市之间的距离 $d=(v_i, v_j)$, 求一条经过 v 中所有城市一次且仅一次的闭合路径 $v_\pi = \{v_{\pi(1)}, v_{\pi(2)}, \dots, v_{\pi(N)}\}$, 使得总行程 $d = \sum_{i=1}^{N-1} d(v_{\pi(i)}, v_{\pi(i+1)}) + d(v_{\pi(N)}, v_{\pi(1)})$ 最小。

1. 抗体编码及亲和度

在求解 TSP 问题中, 抗体 a 对应候选路径, 通常采用实数编码 (以城市的节点序号表示)。抗体群 Ab 可以表示为 $Ab=[a_1, a_2, \dots, a_m]$, m 为抗体的个数。抗体与抗原的亲和度反应抗体与抗原的匹配程度, 一般取求解路径距离的倒数, 为了更好区分抗体间的优劣, 在这里重新定义亲和度函数:

$$\text{affinity}(A_i) = \frac{S_0}{\text{dist}(A_i)} \quad (2.8)$$

式 (2.8) 中 $\text{dist}(A_i)$ 为第 i 个抗体遍历整个城市的距离, S_0 为 TSPLIB 库中提供的已知的最短路径长度。当 $\text{affinity}(A_i)$ 越大时, 表明抗体与抗原的亲和度也越大。

2. 免疫基因操作

T_g^C 主要包括变异, 即 $Z(t)=T_g^C(Y(t))$ 。变异操作可以有多种选择, 针对 TSP 问题, Inver_over 操作算子^[53]在抗体周围的编码空间进行局部搜索, 利用了抗体子种群内部的启发信息, 实验已经证明该操作算子优于传统的交叉算子, 如 PMX、OX、CX 等。3-Opt 变异算子有较强的局部搜索能力, 充分结合 3-Opt 算子与 Inver_over 算子的优势, 以一定概率进行合理分配, 具体见免疫变异操作算子伪代码。在这里免疫基因操作主要包括动态变异概率和免疫变异算子。

(1) 动态变异概率

通过动态地调整两种变异算子的变异概率以尽可能提高算法的收敛速度。动态变异概率如下:

$$p = p \times (1 - t \times \alpha / T) \quad (2.9)$$

式中, p 为变异概率, α 为一般取值很小的常数, t 为当前进化代数, T 为总的进化代数。

(2) 免疫变异算子

免疫变异操作算子伪代码

```

begin:
for  $Y(t)$  中的每一个抗体  $Y_i(t)$  do
{
if (Rand ()  $\leq P$ ) //  $p$  为变异概率
3-OPt ( $Y_i(t)$ );
else
Inver_over ( $Y_i(t)$ );
}
end

```

实验仿真环境: Windows XP 系统, AMD 处理器, 1.90GHz, 448MB 内存, 仿真软件为 MATLAB 7.0。为了更好地说明该算法的有效性, 每个测试实例均测试 30 次, 每次迭代次数设为 TSP 节点数(城市节点数)的 3.5 倍。若算法搜索到全局最优值, 算法也可退出。

实验参数设置: 通过大量的实验数据及由参考文献[8, 130]参数设置方法综合得出, 主要考虑算法的执行时间及求解问题的精度。抗体子种群分为 6 个, 初始变异概率为 0.75, 环境负荷量 $K=75$, 每个子种群抗体初始规模为 35, 克隆算子比例系数 $\alpha=100$, 克隆算子常数 $b=10$ 等参数设置较合理。

2.5.1 三种不同形式 ICA 算法性能比较

为了验证竞争合作型协同进化免疫算法 CCCICA 的有效性, 采用三种不同形式 ICA (文中提出的局部最优免疫优势克隆选择算法) 算法来求解下列 TSP 实例, 实例 1: 对 a280 (城市规模为 280 个, TSPLIB 标准库提供的已知最优值 $S_0=2586.8$) 进行 30 次测试, 三种算法方案设计分别为:

① 无生态种群竞争的纯多种群免疫优势克隆选择算法 (Multi-Population Local Optimization Immunodominance Clonal Selection Algorithm, MICA)。

② 引入环境与生态种群竞争的多种群免疫优势克隆选择算法 (Competitive Coevolutionary Immunodominance Clonal Selection Algorithm, CCICA), 但

高层不做优良个体交叉及优良个体迁移操作。

③ 竞争合作型协同进化免疫优势克隆选择算法 (CCCICA)。

若采用 MICA 算法, 子种群分为 6 个, 每个子种群抗体规模为 35, 子种群抗体变异概率为 0.75, 迭代次数为城市数的 3.5 倍, 30 次测试得到的最差值为 2628.5140, 最优值为 2606.8601, 均值为 2609.1480, 偏离最优路径的平均比率为 0.86%。

若采用 CCICA 算法, 子种群分为 6 个, 每个子种群抗体初始规模为 35, 子种群变异概率为 0.75, 环境负荷量 $K=75$, 种群间的竞争抑制指数为 $a_{ij} = \frac{\alpha \cdot \text{aff}(S_j) + (1-\alpha) \cdot F_{\max}(S_j)}{\alpha \cdot \text{aff}(S_i) + (1-\alpha) \cdot F_{\max}(S_i)}$ ($\alpha < 1$ 为正常数, $\text{aff}(S_j)$ 为第 j 个子种群的平均亲和度, $F_{\max}(S_j)$ 为第 j 个子种群亲和度最大的个体值), 30 次测试得到的最差值为 2626.2182, 最优值为 2597.4407, 均值为 2608.0100, 偏离最优路径的平均比率为 0.82%。

若采用 CCCICA 算法, 子种群分为 6 个, 每个子种群抗体初始规模为 35, 子种群变异概率为 0.75, a_{ij} 及环境负荷量 K 与 CCICA 设置一样, 30 次测试得到的最差值为 2629.0110, 最优值为 2586.7696, 均值为 2603.1524, 偏离最优路径的平均比率为 0.632%。

三种算法比较的收敛曲线如图 2.3 所示。从上述结果及图 2.3 可以看出 CCCICA 的求解精度与收敛速度均优于 MICA 算法及 CCICA 算法。其他实例三种算法的比较如表 2.2 所示。

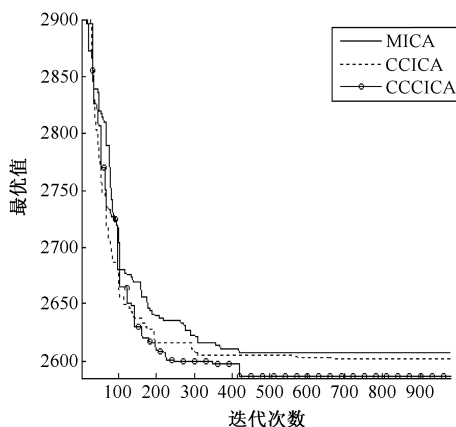


图 2.3 不同算法的收敛 (a280)

表 2.2 其他实例三种算法的比较 (30 次测试)

测试实例(S_0)	Method	Worst	Best	Avg	Std	$T(s)$	Error(%)
Eil51 (428.8718)	MICA	429.1179	428.8718	428.9403	0.11	69.8	0.016
	CCICA	428.9816	428.8718	428.9047	0.04	66.5	0.008
	CCCICA	428.8718	428.8718	428.8718	0	11.48	0
Kroc100 (20750.8)	MICA	20852.2785	20750.7625	20771.0657	40.6	174.3	0.09
	CCICA	20852.2785	20750.7625	20760.9141	30.45	166.8	0.049
	CCCICA	20750.7625	20750.7625	20750.7625	0	132.9	0
Eil101 (642.3)	MICA	650.3956	640.2116	644.4464	3.31	182.3	0.39
	CCICA	649.0493	640.4219	644.2551	2.95	175.8	0.35
	CCCICA	642.2416	640.2116	640.62	0.81	142.9	0
Gr202 (489.099)	MICA	492.0689	488.7174	490.5203	1.78	1615.3	0.29
	CCICA	491.7781	487.8320	489.9422	2.12	1163.4	0.17
	CCCICA	490.3508	487.2631	489.0657	1.73	703.39	0
Pcb442 (5078.35)	MICA	5276.6	5118.0	5193.0	48.95	11554.1	2.26
	CCICA	5214.9945	5113.0214	5178.4137	37.77	15653.1	1.97
	CCCICA	5198.268	5101.513	5111.2	29.02	16653.1	0.65

表 2.2 中的 Best、Worst、Avg、Std、 T 、Error 分别代表一次迭代找到的最优值、最差值、均值、标准方差、平均求解时间及偏离最优路径的平均概率。

$$\text{Error} = \sum_{i=1}^{30} (S_{T_i} - S_0) / 30S_0 \times 100\% \quad (2.10)$$

式中, S_{T_i} 为第 i 次测试得到的值, S_0 为 TSPLIB 提供的已知最优路径值。从图 2.3 及表 2.2 可以看出 CCICA 在求解问题时所得到的最大值、最小值、均值、标准均方差及偏离最优路径的平均概率优于 MICA 算法, CCCICA 所得到的最大值、最小值、均值、标准均方差及偏离最优路径的平均概率均优于 CCICA 算法与 MICA 算法,说明了基于生态种群密度的协同进化算法的有效性,以及优良个体交叉和优良个体迁移操作作用是明显的。同时还可以看出 CCCICA 的求解时间复杂度并没有增大,大部分实例 CCCICA 的求解时间复杂度比 MICA、CCICA 少,这是因为 CCCICA 求解精度高,收敛速度快,能尽快找到最优解或者满意解,所以节约了大量的 CPU 时间。

2.5.2 CCCICA 与其他免疫克隆选择算法的性能比较

实例 2: 对 ch150 (节点城市规模为 150 个) 进行 30 次测试, 各算法的变异初始概率 $P_m=0.75$, 其中 IDIA (文献[8]中提出的基于优势抗体的免疫算法)、ICA (文中提出的局部最优免疫优势克隆选择算法) 抗体种群规模为 150, CCCIDIA (竞争合作型协同进化 IDIA 算法)、CCCICA 算法的抗体子种群数为 6 个, 每个子种群抗体初始规模为 35。比较结果如表 2.3 所示。

表 2.3 其他实例四种算法的比较 (30 次测试)

测试实例(S_0)	Method	Worst	Best	Avg	Std	$T(s)$	Error(%)
Att48 (33524)	IDIA	34076	33524	33666	—	—	0.42
	ICA	33784	3353.7085	33614.85	99.69	22.3	0.27
	CCCIDIA	33588.34	33523.7085	33530.17	19.39	22.5	0.019
	CCCICA	33523.7085	33523.7085	33523.7085	0	11.48	0
Rd100 (7910.4)	IDIA	8182.0	7910.4	8021.4	—	—	1.4
	ICA	7944.3219	7910.3962	7931.0242	12.13	294.12	0.26
	CCCIDIA	7959.09	7910.3962	7922.32	14.43	166.8	0.15
	CCCICA	7911.304	7910.396	7910.67	0.32	142.9	0.003
Ch130 (6110.9)	IDIA	6318.3	6153.7	6218.8	—	—	1.77
	ICA	6231.7526	6112.8669	6170.1797	26.78	288.7	0.97
	CCCIDIA	6170.937	6125.071	6146.81	10.51	295.1	0.59
	CCCICA	6143.75	6110.721	6122.704	7.98	275.0	0.194
a280 (2586.8)	IDIA	2815.2	2621.6	2704.4	—	—	4.55
	ICA	2668.7999	2592.9980	2639.7102	17.51	1756.3	2.05
	CCCIDIA	2626.22	2589.015	2608.01	8.49	1849.6	0.82
	CCCICA	2629.011	2586.769	2603.152	9.8	1601.4	0.63
Pcb442 (5078.35)	IDIA	5468.83	5184.49	5278.40	—	—	3.94
	ICA	5276.6	5118.0	5193.0	48.9	32114.1	2.26
	CCCIDIA	5221.28	5122.23	5157.32	32.1	17653.1	1.55
	CCCICA	5198.268	5101.513	5111.2	29.02	16653.1	0.65
Pr1002 (259068)	IDIA	269499	269343	269355	—	—	3.97
	CCCIDIA	268324.28	266315.33	267320.1	778.3	40165.8	3.19
	CCCICA	265048.12	264768.14	264908.1	108.4	31165.9	2.26

CCCICA 算法获得最大路径距离为 6601.7850,最短路径距离 6530.9025,平均路径距离为 6570.4182, 偏离最优路径值的平均误差率为 0.58%,得到的最短路径长度优于 TSPLIB 标准库所提供的最优路径长度 6532.3, ICA 算法得到最大值 6662.26, 最小值为 6563.16, 平均值为 6607.6, 偏离最优路径的平均概率为 1.15%, CCCIDIA 算法得到的最大值为 6659.9806, 最小值为 6558.16, 平均值为 6597.0687, 偏离最优路径值的平均误差率为 0.99%, 同时 CCCICA 算法也优于文献[8]基于优势抗体的免疫 IDIA 算法获得最大路径长度为 6805.6, 最优路径长度 6568.4, 平均值 6690.2, 偏离最优路径值的误差率为 2.42%, 其收敛曲线如图 2.4 所示。从图 2.4 可以得出 CCCICA 的收敛速度和求解精度均优于 IDIA、ICA、CCCIDIA, 还可以看出 ICA 的收敛速度与求解精度高于 IDIA。这说明局部最优免疫算法优于 IDIA, 但同时 CCCIDIA 的求解精度与收敛速度优于 IDIA, 证实了基于一种竞争合作型的 CCIDIA 算法效果比较好。

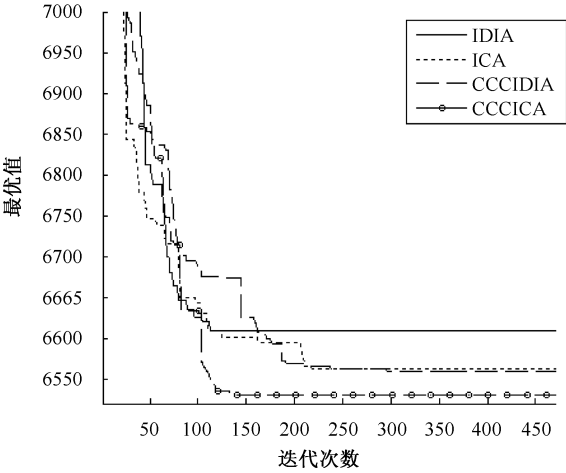


图 2.4 不同算法的收敛 (ch150)

表 2.3 中的 IDIA 为文献[8]基于优秀抗体的免疫算法, ICA 为文中提出的单种群局部最优免疫优势克隆选择算法, CCCIDIA 为竞争合作型协同进化 IDIA 算法。从表 2.3 可以看出, CCCICA 求得的最大值、最小值、均值、标

准方差及偏离最优路径的平均概率优于 IDIA、ICA、CCCIDIA 算法。通过 ICA 算法与 IDIA 算法以及 CCCICA 算法与 CCIDIA 算法的比较证实了局部最优免疫优势算子的高效, 通过 CCCIDIA 算法与 ICA 和 DIA 算法以及 CCCICA 算法与 ICA 算法的比较, 进一步说明了竞争合作型协同进化算法的有效性。

2.5.3 CCCICA 与其他智能算法的性能比较

表 2.4 中 KG (KNIES_TSP_Global)、KL (KNIES_TSP_Local)、KD (KNIES_DECOMPOSE) 的结果引自文献[134], Budinich (Budinich's SOM) 和 ESOM (Expending SOM) 的结果引自文献[135]。ISOM (泛化竞争和局部渗透机制的自组织神经网络方法) 数据结果来自文献[136]。

表 2.4 CCCICA 与神经网络有关方法的比较 (30 次测试)

测试实例	偏离最优路径的平均概率 Error(%)					
	KG	KL	KD	Budinich	ISOM	CCCICA
Eil51	2.86	2.86	3.53	3.10	3.8132	0
St70	2.33	1.51	3.67	1.70	2.1104	0
Eil76	5.48	4.98	6.49	5.32	4.6264	0.288
Rd100	2.62	2.09	4.89	3.16	3.4011	0.003
Lin105	1.29	1.98	2.18	1.71	0.2420	0.017
Pcb442	10.45	11.07	8.00	8.43	8.1955	0.65
Pr1002	7.60	——	7.08	8.75	5.0090	2.31

表 2.5 中 NGA 为带贪婪算子的遗传算法, 其实验数据结果引自文献[8], FFM (填充函数变换方法) 实验数据引自文献[137], ACS (改进的蚁群算法) 优化结果引自文献[138]。从表 2.4 和表 2.5 可知, CCCICA 算法与其他智能算法性能的比较, CCCICA 算法具有明显的优势。随着城市规模的增大, CCCICA 的优势越明显。

表 2.5 CCCICA 与 NGA、ACS、FFM 算法性能的比较 (30 次测试)

测试实例(S_0)	Method	Worst	Best	Avg	Error(%)
Eil51 (428.87)	NGA	460.7428	438.0805	445.7453	3.71
	ACS	437.80	428.87	429.48	0.14
	CCCICA	487.87	428.87	487.87	0
Eil76 (544.37)	NGA	589.30	568.89	577.16	5.84
	ACS	565.16	545.97	553.77	1.73
	CCCICA	550.4968	544.37	546.39	0.288
KroA100 (21285.44)	NGA	23340.8	22924.7	23076.2	8.41
	ACS	22499.74	21285.44	21456.98	0.81
	CCCICA	21636.18	21285.44	21359.60	0.35
D198 (15808.65)	ACS	16936.53	15919.55	16143.66	2.12
	CCCICA	15879.4109	15811.3409	15836.3510	0.18
Att48 (33524)	NGA	35594	34228	34584	3.16
	FFM	33523.71	33523.71	33523.71	0
	CCCICA	33523.7085	33523.7085	33523.7085	0
Rd100 (7910.4)	NGA	8723.1	8291.2	8449.9	6.82
	FFM	7949.32	7910.4	7921.2	0.14
	CCCICA	7911.3049	7910.3962	7910.6688	0.003
Ch130 (6110.9)	NGA	6724.9	6364.8	6570.1	7.51
	FFM	6194.54	6126.51	6152.8	0.69
	CCCICA	6143.7507	6110.7219	6122.7045	0.19
Ch150 (6532.3)	NGA	7374.8	6955.9	7173.6	9.82
	FFM	6560.94	6533.81	6537.3	0.076
	CCCICA	6601.7850	6530.90	6570.4182	0.586
Ch144 (30356)	FFM	30720.94	30355.01	30533.12	0.58
	CCCICA	30643.4098	30355.11	30501.5037	0.48

2.6 本章小结

本章分析了人工免疫算法的特点：人工免疫算法采用基于个体自身亲和度的进化模式，而没有考虑其进化的环境和个体之间的复杂联系对个体进化

的影响。为了提高人工免疫算法的收敛性能,引入生态学中的协同进化思想到人工免疫算法中,充分考虑了环境和子群体间相互竞争的关系,提出了一种竞争合作型协同进化免疫优势克隆选择算法模型(CCCIA),并对算法进行了收敛性分析。通过实验表明,基于生态种群密度的协同进化算法的有效性,优良种群个体免疫杂交与迁移操作算子效果是明显的。通过ICA算法与IDIA算法的比较证实了基于知识演化的局部最优免疫优势算子的高效,通过CCCIDIA算法与ICA和DIA算法的比较,进一步说明了协同进化算法的有效性。针对TSP多个标准实例进行了测试,结果表明:该算法具有较好的收敛性、较高的求解精度和很强的全局搜索能力。但随着问题规模的增大,算法的计算复杂度也会相应增大。

第3章

免疫双态粒子群算法

结合人工免疫系统机理与粒子群的信息交互优点，研究了一种免疫双模态粒子群优化算法。算法将粒子群体分为捕食与探索两种模态，促使粒子群具有动态特性，设计了相关算子，理论上分析了算法的收敛性，并就各参数算子对算法性能的影响进行了分析，通过实验仿真测试了算法的性能。

3.1 引言

粒子群优化算法（Particle Swarm Optimization, PSO）是由 Eberhart 和 Kennedy^[139]两位博士于 1995 年提出来的一种基于群体行为的全局优化进化算法，它源于对鸟类、鱼群等微粒捕食行为的模拟。由于粒子群算法的实现形式比较简单并且优化速度较快，可用于解决大量非线性、不可微和多模态优化问题，现已广泛地应用于科学与工程领域^[140]。然而，大量文献研究表明粒子群算法与其他进化算法类似，同样是基于群体进化，且随着问题规模的增大，算法在进化后期容易陷入早熟收敛，由于这些缺陷限制了粒子群的更广泛应用。因此，学者们致力于从各个方面来研究提高其收敛性能，产生了若干变种粒子群算法。主要改进方式可以归纳为以下两大类。

① 一部分研究人员将其他智能搜索算子引入粒子群算法中，如 Juang 及 Ahmed 将遗传算法中交叉与变异机制融入粒子群算法中，构造一种遗传变异粒子群算法，使粒子群具有变异能力，提高了算法的收敛性能^[141,142]，

同时由于基本遗传算法的变异空间比较固定,对算法性能的提高具有一定的限制性。针对以往经典遗传粒子群算法的缺陷,文献[143]借助小波函数的振幅空间可变与微调能力等优点,提出了一种小波粒子群算法,提高了算法的动态优化性能,但是小波变异幅度较小,算法在收敛速度与求解精度上还有待提高。

② 还有一些研究人员试图改变粒子的运动行为以增强粒子的多样性,例如, Shi 等人提出了一种线性函数与模糊方法来改进粒子群算法模型中的收敛因子,以改善粒子群算法的动态性能^[144,145]。Zhan 等人在文献[146]中提出了一种自适应粒子群算法,该算法通过设计一种精英学习策略及模糊状态进化规则调节机制,提高了算法的收敛速度、全局优化能力及求解精度;但同时该算法的进化状态难以精确确定,将影响算法迭代效率。Liang 等人在文献[147]中提出了一种综合学习型粒子群算法,该算法提供了一种新颖的学习策略,即所有粒子群的个体极值 P_{best} 都存在被选中的更新群体中任何一个粒子速度的可能,研究表明该算法过于注重多样性保持,相对弱化了算法的收敛性与局部精细搜索能力,尤其是求解单模态问题能力不强^[148]。文献[148]提出了一种自适应完全学习粒子群算法(A-CLPSO),通过设计自适应学习算子以提高基本 CLPSO 的优化能力,但该方法并未为粒子群提供逃离局部极值的机制,对提高 CLPSO 的优化性能有限。文献[149]将序列二次规划(SQP)方法引入 CLPSO 算法,提高了 CLPSO 算法的局部优化能力,算法的收敛速度较快,但是算法对连续问题求解精度依然不够高。还有一些学者对微粒的拓扑结构进行了改进,如 Kennedy 和 Mendes 在文献[150]、文献[151]中分别提出了环形、冯诺依曼拓扑结构,改善了粒子群算法寻优性能。这些改进在一定程度上改善了算法性能,然而,早熟收敛的抑制及求解精度的提高依然是 PSO 算法研究的难题,尤其是在复杂高维及多模态优化问题中。

人工免疫系统^[8](Artificial Immune System, AIS)是模拟生物免疫系统的机理而构造的高性能、自组织、鲁棒性好的人工智能系统。其免疫克隆选择机制中存在着克隆、超变异、抗体与抗原特性结合,记忆细胞产生等过程,在保证收敛速度的同时,又能够维持抗体的多样性;其免疫受体编辑机制中允许细胞受体在特定的条件下还可以发生重组或突变,原有的抗原受体特异性向其他特异性漂移突变等,更进一步丰富了抗原受体的多样性。免疫机制

能有效改善其他进化算法的寻优性能。

分析了基本粒子群算法易陷入局部极值的原因,标准粒子群的状态主要集中在收敛状态,一旦最优粒子陷入局部最优值,其邻域微粒都随着陷入局部点。结合鸟类、鱼类等微粒捕食行为及人工免疫系统的特点,本章提出一种免疫双态粒子群算法,在搜索的过程中依据群体进化因子协调群体勘探与开采,把粒子群划分为“捕食”与“探索”两种模态,以扩大解的搜索空间,有效抑制了收敛过程中的早熟停滞现象,对处于“捕食”状态的精英微粒采用增强学习策略,加快了算法的收敛速度。引入免疫系统的克隆选择和受体编辑机制,增强群体逃离局部极值能力,提高群体的多样性分布及对多模态问题极值的全局寻优能力。经典测试函数实验仿真表明,该算法收敛速度快,求解精度高,种群多样性好。

3.2 粒子群算法优化原理

粒子群优化算法^[139]是由 Kennedy 等于 1995 年提出的一种模拟鸟群等微粒觅食行为的进化计算方法,该算法通过个体之间的协作来进行迭代优化。设粒子群体规模为 N ,每个粒子在 D 维搜索空间中,记粒子($i=1,2,\dots,N$)的当前位置是 $X_i = \{X_{i1}, X_{i2}, \dots, X_{id}\}$,飞行的速度为 $V_i = \{V_{i1}, V_{i2}, \dots, V_{id}\}$ 每个粒子当前找到的最优解为 $Pbest_{id}$ 种群当前找到的全局最优点为 $Gbest_d$ 。

$$V_{id}(t+1) = \omega V_{id} + c_1 * rand_1() (Pbest_{id}(t) - X_{id}(t)) + c_2 * rand_2() (Gbest_d(t) - X_{id}(t)) \quad (3.1)$$

其中, ω 为惯性权重,一般随着进化线性减少; c_1 、 c_2 为调节 $Pbest_{id}$ 和 $Gbest_d$ 相对重要性的参数; $rand()$ 生成介于 0 和 1 之间的随机数。这样,可以得到粒子的下一位置:

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (3.2)$$

从微粒运动数学表达式(3.2)可知,当群体微粒趋同时,表达式后两项社会认知部分($Gbest_d - X_{id}(t)$)与个体自身认知部分($Pbest_{id}(t) - X_{id}(t)$)趋于 0 时,而 ω 小于 1,其速度将迅速下降到 0,粒子群将停止运动。基本粒子运动轨迹图如图 3.1 所示。

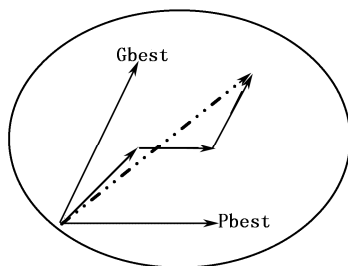


图 3.1 基本粒子运动轨迹图

每个粒子的优劣程度根据适应度函数来评价，这与待解决的具体问题相关。PSO 算法的基本流程如图 3.2 所示。

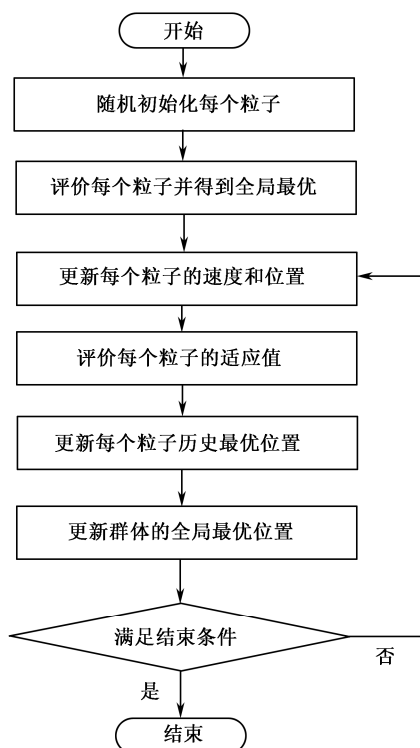


图 3.2 PSO 算法的基本流程

粒子群算法的基本流程如下。

步骤 1: 在 n 维搜索空间中, 随机初始化粒子群粒子的速度与位置。

步骤 2: 根据目标函数评价每个微粒当前位置的适应值。

步骤 3: 对每个微粒, 将其当前获得的适应度值与所经历过的个体最优位置 P_{best} 的适应值进行比较, 若较好, 则将其作为当前的 P_{best} 值。

步骤 4: 对每个微粒, 将其当前获得的适应度值与群体全局所经历过的最优位置 G_{best} 的适应度值进行比较, 若较好, 则将其作为当前全局的 G_{best} 值。

步骤 5: 根据式 (3.1)、式 (3.2) 对每个微粒的速度、位置进行更新。

步骤 6: 判断是否满足终止条件, 若满足则输出最优解; 否则, 返回到步骤 2 重新计算。

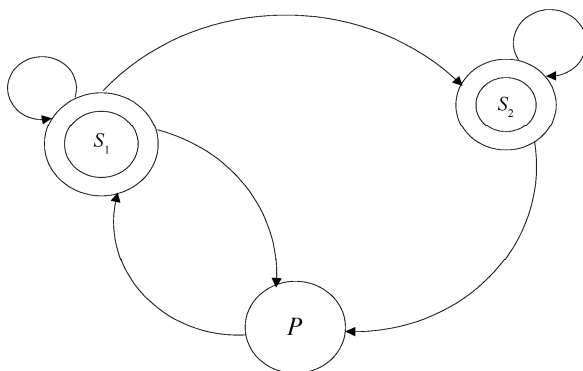
3.3 免疫双态粒子群算法

3.3.1 双态粒子群算法机理

由基本微粒运动轨迹方程可知, 基本粒子群优化算法微粒在搜索中总是沿着前一代速度方向飞行, 运行速度很快。但是, 算法运行过程中, 如果某微粒发现一个当前最优位置, 邻域内其他微粒将迅速向其靠拢。如果该最优位置为一局部最优点, 粒子群就无法在解空间内重新搜索, 因此, 容易陷入局部极值, 而基本粒子群没有提供逃离局部最优解的机制, 因此解决 PSO 算法中的停滞现象和保证其具有较强的探索能力的方法就是在“探索”和“利用”之间建立一个平衡态。既要使算法的搜索空间尽可能大, 以寻找那些可能存在最优解的解空间, 同时也要充分利用微粒搜索过程中遗留的有效历史信息, 使算法搜索的侧重点放在那些可能具有最优解的空间, 从而以更大的概率收敛到全局最优解。对鱼类、鸟群等微粒捕食行为过程的研究表明, 真实群体智能社会中的群居智能体各司其职, 又相互依赖协作, 形成一个有机的整体, 从中得到启迪, 在群体进化的过程中, 根据群体进化因子, 可以将具有 N 只粒子群分成两种不同模态行为特征子群, 即探索群体与捕食群体。

探索群体有潜力探索有希望的区域以寻求更优的解, 同时又不失群体的

多样性, 捕食群体有能力在已探索到的最优区域进行精细搜索, 探索群体与捕食群体的比例设为 $r_2:r_1$ 。这样对于整个粒子群来说, 不会因为遇到局部极值点而停止搜索, 群体在整个搜索过程中都会有充足的动量在解空间中进行搜索。其群体行为可以用简单的自动机模型表示, 如图 3.3 所示。



S_1 为捕食状态, S_2 为探索状态, P 为群体

图 3.3 双模态粒子群自动机模型

1. 捕食状态

鸟类、鱼类等微粒在群体中比单独行动时能更多更快地找到食物。如果粒子群中的一个微粒找到了食物, 其他成员也可以捕食。在捕食状态下粒子群行为与传统的粒子群算法一致。设粒子群体规模为 N , 每个粒子在 d 维搜索空间中, 记粒子 $i(i=1,2,\dots,N)$ 的当前位置是 $x_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}^T$, 飞行的速度为 $v_i = \{v_{i1}, v_{i2}, \dots, v_{id}\}^T$, 每个粒子当前找到的最优解为 P_{id} , 种群当前找到的全局最优点为 P_{gd} 。为了减少进化迭代过程中粒子离开搜索空间的可能性, 通常限定 $V_{\min} \leq V_{id} \leq V_{\max}$ 。

$$V_{id}(t+1) = \omega V_{id} + c_1 * \text{rand}() (P_{id}(t) - X_{id}(t)) + c_2 * \text{rand}() (P_{gd}(t) - X_{id}(t)) \quad (3.3)$$

ω 为惯性权重。一般随着进化线性减少, 如下式:

$$\omega_t = \omega_{\text{start}} - (\omega_{\text{start}} - \omega_{\text{end}}) \cdot \frac{t}{T} \quad (3.4)$$

其中, T 为群体的最大迭代次数, t 为当前的迭代代数, ω_{start} 、 ω_{end} 分别是初始惯性权重的最大值和最小值。 c_1 、 c_2 为调节 P_{id} 和 P_{gd} 相对重要性的参数, $\text{rand}()$ 生成介于 0 和 1 之间的随机数。这样, 可以得到粒子的下一位置:

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (3.5)$$

2. 探索状态

在优化过程中, 粒子群根据当前群体所处的状态, 经过有限次的探寻, 以寻找更优的状态, 如果没有找到更优的解, 这说明整个群体已陷入局部极值点, 群体中一部分微粒将被分散, 转而去探索新的解空间。这时依据进化因子就要分散一部分粒子群转为探索状态。当微粒从捕食状态转化为探索状态时, 微粒将重新随机初始化分布在搜索空间中。

$$X_{kd}(t) = X_{kd}^{\min} + \text{rand}() * (X_{kd}^{\max} - X_{kd}^{\min}) \quad (3.6)$$

其速度位置更新公式如下:

$$V_{kd}(t+1) = \omega * \text{sign}(r) * V_{kd} + c_1 * \text{rand}() * (P_{kd}(t) - X_{kd}(t)) + c_2 * \text{rand}() * (P_{gd}(t) - X_{kd}(t)) + c_3 * u(0,1) * (X_{kd}(t) - hP_{gd}) \quad (3.7)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (3.8)$$

$$\text{sign}(r) = \begin{cases} 1, & r \geq 0.1 \\ -1, & r < 0.1 \end{cases} \quad (3.9)$$

其中, c_3 为(0,1)之间的随机数, r 为随机数, $\text{sign}(r)$ 为符号函数, 为了寻求更优目标, 微粒可能改变飞行方向^[152], $u(0,1)$ 为高斯分布函数。 hP_{gd} 为当前探索种群中最优粒子在第 d 维的位置。为了继承种群中历史最优粒子的搜索优势, 一部分精英粒子不参与探索。双模态动态行为表明对于整个粒子群来说, 不会因为遇到局部极值点而停止搜索, 群体在整个搜索过程中都会有充足的动量在搜索空间中进行搜索, 找到全局最优值的概率大为提高。

种群进化加速因子定义如下式所示。

$$f = \frac{\text{Gbest}_i - \text{Gbest}_{i-1}}{\text{Gbest}_{i-1} - \text{Gbest}_{i-2} + \alpha} \quad (3.10)$$

Gbest_i 为第 i 代种群最优适应度值, $\alpha \in (0,1)$ 为平滑系数, 防止分母为 0。式 (3.10) 表明, 当 $f=0$ 时进化停止; 当进化因子 $f>0$ 时, 表明群体一直处于进化状态; 当 $f>1$ 时进化加速; 当 $f<1$ 时群体进化速度减慢, 这时如果群

体寻优结果没达到最优适应度值, 群体就要分出一部分粒子群转为探索状态以寻求更大解空间。

当 $f < 1$ 时, 以一定的概率进行探索, if $\text{rand} < (1-f)$ then 以一定比例的粒子进行探索。

3.3.2 精英粒子局部增强学习算子

精英粒子为群体领导者, 对群体整体进化有着重要的推动作用。为防止精英微粒陷入局部极值点, 精英微粒设计一种局部增强学习算子以协助部分精英微粒跳出局部极值点:

$$G_{kd} = \text{Gbest} + m \cdot (X_{r1d} - X_{r2d}) \cdot (1 - \frac{t}{T}) \quad (3.11)$$

式中, G_{kd} 为增强后的微粒, Gbest 为当前代全局最优个体, m 为限幅常数, t 为代数, T 为算法执行总代数, 且 $r_1 \neq r_2$, x_{r1d} 、 x_{r2d} 随机选取。增强算子促使这部分微粒在当前种群中的最优个体附近进行随机微调便于逃离局部点。增强算子可以用来增强算法局部搜索能力, 而精英粒子的最优信息不断向种群其他微粒传播加快了算法的收敛速度。

3.3.3 粒子免疫优化

研究表明多数粒子在进化后期易集中收敛于局部极值点, 失去多样性, 且粒子自身没有适应性变异能力, 致使群体后期优化能力下降, 很难找到全局最优极值点。引入人工免疫系统^[22]中的克隆选择机制和受体编辑机制, 可以增强粒子群多样性、大范围变异能力及增大收敛速度。

1. 个体极值免疫克隆选择

步骤 1: 各个粒子的个体极值 $\{P_{1d}, P_{2d}, \dots, P_{Nd}\}$ 组成一个临时的克隆种群。将临时克隆种群每一粒子视为抗体, 选择 $k(k < N)$ 个最好的抗体个体组成子群体 Sub_1 , 对子群体 Sub_1 进行克隆复制, 公式如下:

$$\text{Sub}_1(i) = \text{round}(\frac{\beta N}{i} + b) \quad (3.12)$$

其中 N 为种群规模, $\beta \in (0,1)$ 。保证每个抗体都有一定克隆数量, 故加上了常量 $b \geq 1$ 的整数。经过克隆扩增生成新群体 Sub_2 。

步骤 2: 对群体 Sub_2 中的每个个体实施高频变异, 其方法为基于时间变尺度的邻域内变异算子, 其变异公式如下:

$$P_{id}^{\text{new}} = P_{id} + [(r_d > P_m)] * \eta * P_{id} * U(0,1) - [(r_d \leq P_m)] * \eta * P_{id} * U(0,1) \quad (3.13)$$

$$[r_d > P_m] = \begin{cases} 1, & r_d > P_m \\ 0, & \text{其他} \end{cases} \quad (3.14)$$

其中, r_d 为随机数, $P_m = 0.5$ 。

$$\eta(t) = 1 - r^{[1-(t/T)]^b} \quad (3.15)$$

其中, t 是群体当前进化代数, T 为总的进化代数, b 为正常数, 一般取值为 2, $r \in (0,1)$ 、 $U(0,1)$ 为 $(0,1)$ 之间的均匀随机数。从式 (3.15) 可看出, 在进化初期, 对于较小的 r 值时, $\eta(t) \approx 1$, 此时的变异空间大; 而在进化后期, t 接近 T 时 $\eta(t) \approx 0$, 在小范围空间内进行局部非线性搜索。

步骤 3: 免疫选择操作。从克隆变异后的个体中选择亲和度最优的个体进入下一代, 即 $S_i(k) = T_s^C(Z(K)) = \max \{\text{affinity}(Z_i(k))\}$, 然后, $S_i(k) \cup A_i(k) \rightarrow A_i(k+1)$, 这样通过局部择优实现了种群的映射压缩, 同时保证了抗体群中的最优解不会变差。

2. 受体编辑

受体编辑机制^[7]是指生物免疫系统中 T 细胞和 B 细胞受体在特定的条件下其基因位结构发生新的重排或突变, 使其受体特异性发生改变, 其原有的抗原受体特异性向其他特异性漂移从而导致亲和力变化。采用转基因动物模型进行的体内受体编辑的医学实验研究表明, 大约 25% 的 B 细胞发生了受体编辑, 而受体编辑进一步丰富了抗原受体的多样性, 在算法执行过程中, 每隔一定的代数就要鉴别出不活跃的、趋向凋亡的细胞受体, 即对不活跃的微粒集合中的 25%~30% 进行受体编辑。

$$\text{aff}_{\text{avg}} = \frac{1}{n} \sum_{i=1}^n \text{aff}_i \quad (3.16)$$

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (\text{aff}_i - \text{aff}_{\text{avg}})^2 \quad (3.17)$$

式 (3.16) 中 aff_{avg} 为群体中的平均亲和度； σ^2 为群体适应度方差，反映了群体的收敛程度，该式数值越小，则粒子群趋于收敛。当适应度方差为趋于零的一个很小的阈值时，而算法得到的最优解还未达到为理论全局最优解或期望最优解，此时认为群体已陷入局部最优解。对于满足不等式 (3.18) 的抗原受体 i 进行受体编辑。

$$\frac{\text{aff}_i - \text{aff}_{\min}}{\text{aff}_{\max} - \text{aff}_{\min} + \varepsilon} < \theta \quad (3.18)$$

式 (3.18) 中 ε 为 (0,1) 之间的平滑系数，以保证分母不为 0， θ 为阈值。

对于不活跃的、趋向凋亡的细胞受体，利用非线性 Logistic 序列进行随机又有规律的位置大漂移，增强抗体的多样性。受体编辑具体如下式：

$$X_{id}' = X_{id} + \frac{(X_{id}^{\max} - X_{id}^{\min})}{m} U_{r+1} \quad (3.19)$$

其中， X_{id}^{\max} 、 X_{id}^{\min} 为位置的上界、下界， m 为正常数，视具体问题而定。

U_{r+1} 为混沌序 Logistic，其中 U_{r+1} 为混沌序列 Logistic^[153] 映射如下：

$$U_{r+1} = \mu U_r (1 - U_r) \quad (3.20)$$

其中， $r=0,1,2,\dots$ ， $0 < U_0 < 1$ ； μ 为系统的状态控制参量，已证明^[153]，当 $\mu=4$ 且初始值 $U_0 \notin (0.25, 0.5, 0.75, 1)$ 时，式 (3.20) 中系统完全处于混沌状态， U_{r+1} 在 (0,1) 范围内遍历。

3.4 IBPSO 算法流程及性能分析

3.4.1 算法 IBPSO 流程

免疫双态粒子群算法 (IBPSO) 步骤描述如下。

步骤 1: 随机初始化 m 个微粒的位置与速度，以及相关参数。

步骤 2: 评价各微粒的初始适应值，保存初始最好位置及初始最优适应值。

while(不满足退出条件)do //退出条件为找到相应的全局最优值或达到设定的截止代数。

步骤 3: for 对粒子群中每只粒子 do。

根据式 (3.3)、式 (3.5) 更新粒子的位置与速度，计算各粒子适应度值。

计算群体进化加速因子 f 。

if ($f < 1$) & 适应度值没达到全局最优值)

```
{      if( $u(0,1)<1-f$ )      then
```

将粒子群的适应度值排序，性能较好的 $m*r_1$ 按状态 1 行动，其中捕食状态中的部分优秀粒子进行增强学习；

其余 $m*r_2$ 只粒子按状态 2 行动，探索种群按照式 (3.6)、式 (3.7)

更新粒子的位置与速度。

```
}
```

```
end if
```

如果各粒子适应度优于相应粒子历史最优适应度，相应粒子 P_{id} 更新。

如果最优粒子适应度优于历史最优粒子适应度， P_{gd} 更新。

```
end for
```

步骤 4：各个粒子的个体极值 P_{id} 生成一个临时的克隆种群，进行克隆扩增。

步骤 5：对临时克隆种群基于变尺度邻域内变异，免疫选择操作。

步骤 6：对趋于凋亡微粒进行受体编辑。

```
end while
```

步骤 7：输出结果，算法运行结束。

3.4.2 算法性能及收敛性分析

1. 算法性能分析

IBPSO 算法在进化初期，群体搜索能力较强，都聚集在捕食状态下集中搜索。随着迭代的进行，寻优速度减慢，依据进化加速因子，群体被分为两种模态，让一部分微粒转为探索状态，此时解空间进一步扩大，有利于寻找优良解区域。对处于捕食状态的精英粒子进行有指导性的局部增强学习，优秀粒子能正确导向其他微粒的飞行，信息通过邻域传播，进而加速群体收敛速度。基于个体极值的克隆选择机制，扩大抗体的分布性，进一步提高全局搜索能力，有利于多模态（多峰）优化问题求解。随着进化的进行，群体微粒之间的距离逐渐聚集，种群失去多样性。这时受体编辑机制非线性 Logistic 序列进行随机又有规律的位置大漂移，在进化整个过程中保持了种群微粒多样性，促使微粒有充足的动量在解空间中搜索，提高了微粒的全局搜索能力。

综上所述，IBPSO 算法有着广阔的解空间、高效的精细搜索策略、超强的逃离局部极值机制及多样性保持机制，因此该算法具有收敛速度快和求解精度高的特点。

2. 收敛性分析

假设：① 问题 P 的定义域 Ω 为 R^n 的有界闭区域；

② 目标函数 $f(x)$ 是区域 Ω 上的连续函数。

定理 3.1: 设 $\{x(k)\}$ 是由 IBPSO 算法产生的种群序列，其中 $x^*(k) \in X(k)$ 为第 k 代种群中的最优个体，即 $x^*(k) = \operatorname{argmin}_i f(x_i(k))$ ，如果问题 P 中的目标函数和可行域满足假设 1，则有 $p\{\lim_{k \rightarrow \infty} f(x^*(k)) = f^*\} = 1$ ，即种群序列以概率 1 收敛于全局最优解。

证明：① 处于捕食状态的微粒前后两代的位移变化量变为

$$\Delta x_i(t) = \omega V_{id} + c_1 * \operatorname{rand}() (P_{id}(t) - X_{id}(t)) + c_2 * \operatorname{rand}() (P_{gd}(t) - X_{id}(t))$$

设： $\phi_0 = \omega V_{id}$, $\phi_1 = c_1 * (P_{id}(t) - X_{id}(t))$, $\phi_2 = c_2 * (P_{gd}(t) - X_{id}(t))$

$$\Delta x_i(t) = \phi_0 + \phi_1 * r_1 + \phi_2 * r_2, \quad r_1, r_2 \sim N(0, \sigma_i)$$

② 处于探索状态的微粒种群前后两代的位移变化量变为

$$\Delta x_k(t) = \omega * \operatorname{sign}(r) * V_{kd} + c_1 * \operatorname{rand}() (P_{kd}(t) - X_{kd}(t)) + c_2 * \operatorname{rand}() (P_{gd}(t) - X_{kd}(t)) + c_3 * u(0, 1) (X_{kd}(t) - hP_{gd})$$

设： $\phi_0 = \omega * \operatorname{sign}(r) * V_{kd}$, $\phi_1 = c_1 * (P_{kd}(t) - X_{kd}(t))$

$$\phi_2 = c_2 * (P_{gd}(t) - X_{kd}(t)), \phi_3 = c_3 * (X_{kd}(t) - hP_{gd})$$

$$\Delta x_k(t) = \phi_0 + \phi_1 * r_1 + \phi_2 * r_2 + \phi_3 * r_3, \quad r_1, r_2, r_3 \sim N(0, \sigma_i)$$

③ 于是双态粒子群位移变化量可以用下式表示：

$$\Delta x_i(t) = \phi_0 + \phi_1 * r_1 + \phi_2 * r_2 + \psi * \phi_3 * r_3, \quad \text{其中 } \psi = \begin{cases} 1, & \text{if explore} \\ 0, & \text{其他} \end{cases}$$

上述中 explore 表示群体处于探索状态，显然有

$$\Delta x_i(t) \sim N(0, \sigma_i)$$

由于克隆变异选择操作之后获得的最优解至少比操作之前的要好，以及精英粒子局部增强学习采用最优选择策略。因此，最优种群最优信息不会退化。由文献[154,155]可知，定理 3.1 显然成立（详细证明参考文献[154,155]）。

3.5 仿真实验及分析

为了验证本章提出的免疫双态粒子群算法 (IBPSO) 在收敛速度、全局优化性能、多模态寻优能力及高维问题上的求解性能, 引入基准优化测试问题, 进行分析, 将该算法与其他算法变种进行了对比。这些基准函数具有不同的特点, 可以充分考察新型算法对不同类型问题的优化性能。

在表 3.1 中, 测试函数形态各异, 都是难度较大的复杂优化问题, 具有很好的测试性能, $f_1 \sim f_5$ 为单模态函数, $f_6 \sim f_{10}$ 为复杂非线性多模态函数, 可以用于测试算法收敛速度及精度。这些函数都有着典型的特点, 如 f_2 (Schwefel's) 函数是一个非连续的、单模态函数, 可以分析算法的执行性能, f_4 (Rosenbrock) 函数是一个经典复杂优化问题, 它的全局最优点位于一个平滑、狭长的抛物线形山谷内。由于函数仅仅为优化算法提供了少量信息, 使算法很难辨别搜索方向, 找到全局最小点的机会微乎其微。 f_8 (Ackley) 局部极值点较多且距全局最优点较远, 其寻优难度大, f_{10} (Generalized Penalized) 函数局部极值点无穷多, 且惩罚扰动较大, 收敛到理想值难度极大。表 3.1 中测试函数来自文献[146]。

实验设计: 本章所有的实验在 AMD Athlon(tm) II X2 250 处理器, 主频 3.0GHz, 3.25GB 内存, Windows XP2, Visual C++ 6.0 软件平台上进行。为了减少实验统计误差, 所有实例独立运行 30 次, 最大迭代次数为 2000 代, 对平均统计结果进行比较。文中 HPSOM 算法表示基于随机均匀变异的粒子群算法^[156], HGAPSO 为基于遗传算法的混合粒子群算法^[141], HPSOWM 为基于小波变异的粒子群算法^[143], CLPSO 为全面学习粒子群算法^[147], A-CLPSO 为自适应全面学习粒子群算法^[148], APSO 表示一种自适应模糊分类粒子群算法^[146]。实验参数设置: HPSOM、HGAPSO、HPSOWM 参数设置参照文献[143]。CLPSO 算法中 $\omega \in [0.9, 0.4]$, $c_1=c_2=1.49445$; APSO 算法中 $\omega \in [0.9, 0.4]$, $c_1=c_2=2.0$ 。为了保证测试公平, 所有粒子群算法单个种群粒子规模为 50, 文中 IBPSO 算法 $\omega \in [0.91, 0.4]$, $c_1=c_2=1.49445$ 。IBPSO 算法克隆变异初始概率为 0.7。

表 3.1 测试函数

测试函数	区间范围	最优值
$f_1(x) = \sum_{i=1}^D x_i^2$	$-100 \leq x_i \leq 100$	0
$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$-10 \leq x_i \leq 10$	0
$f_3(x) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$	$-100 \leq x_j \leq 100$	0
$f_4(x) = \sum_{i=1}^D (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$-10 \leq x_i \leq 10$	0
$f_5(x) = \sum_{i=1}^D ix_i^4 + \text{random}(0,1)$	$-1.28 \leq x_i \leq 1.28$	0
$f_6(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$-500 \leq x_i \leq 500$	0
$f_7(x) = \sum_{i=1}^D [y_i^2 - 10 \cos(2\pi x_i) + 10]$ where, $y_i = \begin{cases} x_i, x_i < 0.5 \\ \text{round}(2x_i)/2, x_i \geq 0.5 \end{cases}$	$-5.12 \leq x_i \leq 5.12$	0
$f_8(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i}) - \exp[\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)] + 20 + e$	$-32 \leq x_i \leq 32$	0
$f_9(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	$-600 \leq x_i \leq 600$	0
$f_{10}(x) = \frac{\pi}{D} \{ 10 \sin(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \}$ $+ \sum_{i=1}^D u(x_i, 10, 100, 4),$ where $y_i = 1 + \frac{1}{4}(x_i + 1), u(x_i, a, k, m) = \begin{cases} k(xi - a)^m, x_i > a \\ 0, -a \leq x_i \leq a \\ k(-x_i - a)^m, x_i < -a \end{cases}$	$-50 \leq x_i \leq 50$	0

3.5.1 BPSO（双模态粒子群）比例参数

为了分析探索微粒与捕食微粒的比例对($r_2:r_1$)双态粒子群算法性能的影响,选取了实例 f_2 (Schwefel's)函数(30维,理论最优值 $f_{\min}=0$)和 f_{10} (Generalized Penalized)函数(30维,理论最优值 $f_{\min}=0$)进行测试,如表3.2和表3.3所示。其中, Best、Worst、Mean、Std·Dev表示测试获得的最优值、最差值、平均值、标准方差。

表 3.2 参数 $r_2:r_1$ 对 BPSO 算法的影响 (f_2 , 30 次测试)

$r_2:r_1$	Best	Worst	Mean	Std·Dev
0:1	0.086	1.59	0.69	0.41
0.1:0.9	1.55e-3	2.0e-2	6.81e-3	5.9e-3
0.2:0.8	1.91e-4	4.65e-3	2.1e-3	1.59e-3

续表

$r_2:r_1$	Best	Worst	Mean	Std • Dev
0.3:0.7	1.54e-4	1.01e-2	1.9e-3	2.36e-3
0.5:0.5	5.98e-5	2.02e-3	6.46e-4	4.89e-4
0.6:0.4	3.35e-5	2.65e-3	4.54e-4	6.11e-4
0.7:0.3	1.52e-5	1.49e-3	2.51e-4	2.99e-4
0.75:0.25	1.27e-5	2.01e-4	7.81e-5	6.31e-5
0.8:0.2	1.61e-5	2.36e-3	2.82e-4	4.76e-4
0.85:0.15	1.37e-5	1.09e-3	1.95e-4	2.64e-6
0.9:0.1	2.61e-5	1.99e-3	2.64e-4	4.13e-4
1:0	2.16e-5	1.69e-3	2.46e-4	4.87e-4

表 3.3 参数 $r_2:r_1$ 对 BPSO 算法的 (f_{10} , 30 次测试)

$r_2:r_1$	Best	Worst	Mean	Std • Dev
0:1	4.26	23.40	13.46	7.73
0.1:0.9	1.69	24.77	10.63	7.28
0.2:0.8	0.502	19.52	9.901	6.30
0.3:0.7	0.23	24.88	9.52	7.80
0.5:0.5	0.11	24.50	7.71	7.52
0.6:0.4	3.59e-14	15.98	7.11	5.05
0.7:0.3	1.63e-14	15.45	6.59	5.90
0.75:0.25	5.62e-15	0.78	0.173	0.296
0.8:0.2	1.41e-14	12.34	4.37	4.29
0.85:0.15	1.20e-14	17.92	7.39	6.69
0.9:0.1	9.57e-2	29.19	9.54	7.44
1:0	0.11	34.53	13.82	8.42

从表 3.2、表 3.3 可以得出当 $r_2:r_1=0.75:0.25$ 时, 即当探索群体与捕食群体比例($r_2:r_1$)为(0.75:0.25)时算法的性能较好, 最优值、最差值、均值以及标准均方差优于其他比例。之所以探索群体的数量为捕食群体数量的 3 倍左右, 是因为须探索的解空间比局部精细搜索解的空间大得多。 $r_2:r_1$ 的比例起到了协调平衡算法探索与利用的作用。当 $r_2:r_1$ 过大时, 算法的利用

能力（局部精细搜索）不足而探索能力过剩；当 $r_2:r_1$ 过小时，算法的利用能力过剩而探索能力不足。表中 $r_2:r_1$ 为(0:1)表明算法完全处于捕食收敛状态， $r_2:r_1$ 为(1:0)时表明微粒完全处于探索状态，从实验结果可知算法完全处于捕食收敛状态，或者完全处于探索状态时算法收敛性能相对较差。实验结果证明保留性能最好的 25%左右的微粒在捕食状态，其余 75%的微粒转为探索状态搜索其算法收敛性能最佳。

3.5.2 算法精度比较

求解精度是衡量算法的一个重要指标，表 3.4 中表述了算法收敛精度与标准方差，图 3.4 表述了 10 个评价函数 7 种变种 PSO 算法的收敛曲线。表 3.4 中 Mean、Std.Dev 分别表示适应度的均值、标准方差。从表 3.4 中可以看出在 $f_1 \sim f_7$ 及 f_{10} 函数问题上，IBPSO 算法的均值与标准方差明显优于其他 6 种变种粒子群算法，在 f_8 问题上 IBPSO 算法在均值与方差方面优于 HPSOM、HGAPSO、HPSOWM、CLPSO、APSO 算法；同时在 f_8 问题上 IBPSO 的均值优于 A-CLPSO 算法的均值，两算法的标准均方差相等。在 f_{10} 问题上 IBPSO 算法在均值与方差方面优于 HPSOM、HGAPSO、HPSOWM、CLPSO、APSO 算法，同时在 f_{10} 问题上均值、标准均方差方面 IBPSO 算法与 A-CLPSO 算法相等。

表 3.4 IBPSO 与其他 6 种改进 PSO 方法的比较（30 维）

Function		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	IBPSO
f_1	Mean	7.25e-07	3.32e-12	1.65e-8	7.91e-10	1.83e-57	1.02e-42	6.58e-120
	Std.Dev	1.81e-6	6.40e-12	4.09e-8	3.21e-10	3.9243e-057	1.79e-42	3.54e-119
f_2	Mean	7.69	6.33	1.07e-3	3.94e-6	7.11e-33	1.54e-27	1.38e-68
	Std.Dev	6.66	9.12	1.96e-3	8.15e-7	1.49e-33	1.14e-26	1.63e-68
f_3	Mean	9516.23	3451.45	4.72	15371.6	22.0	0.275	2.28e-135
	Std.Dev	5605.6	3716.33	4.93	2812.43	209.67	0.247	1.02e-134
f_4	Mean	1812.03	39.49	37.09	29.88	28.03	19.04	7.10e-12
	Std.Dev	3680.74	34.53	25.02	9.56	14.48	12.29	2.81e-11
f_5	Mean	0.188	0.012	3.58e-3	2.4e-2	2.45e-3	9.84e-3	8.19e-5
	Std.Dev	0.96	0.005	1.18e-3	5.2e-3	6.93e-4	2.72e-3	9.2e-5

续表

Function		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	IBPSO
f_6	Mean	93.93	68.75	58.70	69.33	2.59	4.01	3.32e-2
	Std.Dev	32.78	23.08	18.39	6.21	1.51	1.50	0.179
f_7	Mean	99.83	49.33	48.33	56.46	10.32	3.33e-2	0
	Std.Dev	23.35	27.74	16.19	6.69	1.899	0.1795	0
f_8	Mean	1.91	3.57e-4	0.836	16.77	4.14e-15	0.05	5.89e-16
	Std.Dev	3.41	1.07e-3	0.736	3.06	0	0.2696	0
f_9	Mean	9.05	0.035	1.34e-2	5.26e-6	0	0.018	0
	Std.Dev	27.11	0.036	1.29e-2	1.57e-5	0	0.0153	0
f_{10}	Mean	0.746	0.010	0.318	1.03e-8	9.81e-30	0.0172	1.57e-32
	Std.Dev	0.956	0.031	0.613	6.59e-9	5.28e-29	0.054	2.74e-48

从表 3.4 中数据及图 3.4 可以看出 IBPSO 性能明显优于其他 6 种变种 PSO，其原因是双态免疫粒子群算法（IBPSO）在迭代过程中，精英粒子不断地学习迫使最优适应度值不断发生变化以及克隆选择基于变尺度邻域内变异算子，当陷入局部极值时，也具有微变能力逃离局部极值正确导向其他微粒的飞行；探索群体利用更大概率找到解空间的全局最优值。受体编辑机制对于不活跃的、趋向凋亡的细胞受体，利用非线性 Logistic 序列进行随机又有规律的位置大漂移，保持了群体多样性。

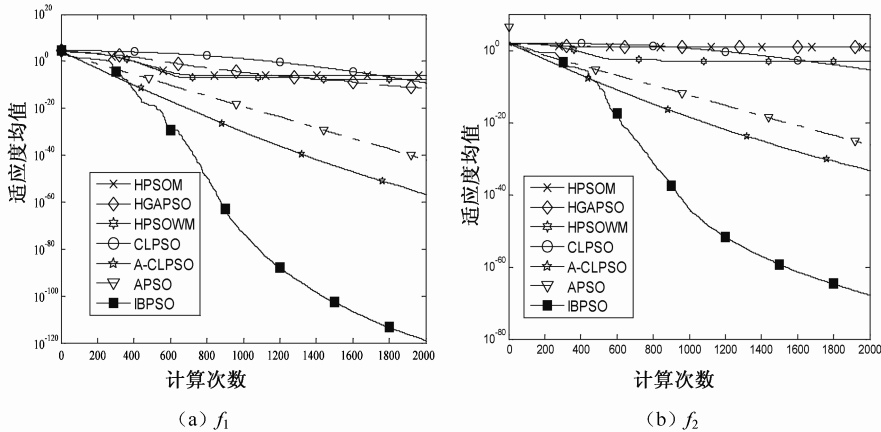


图 3.4 7 种变种粒子群算法收敛曲线比较

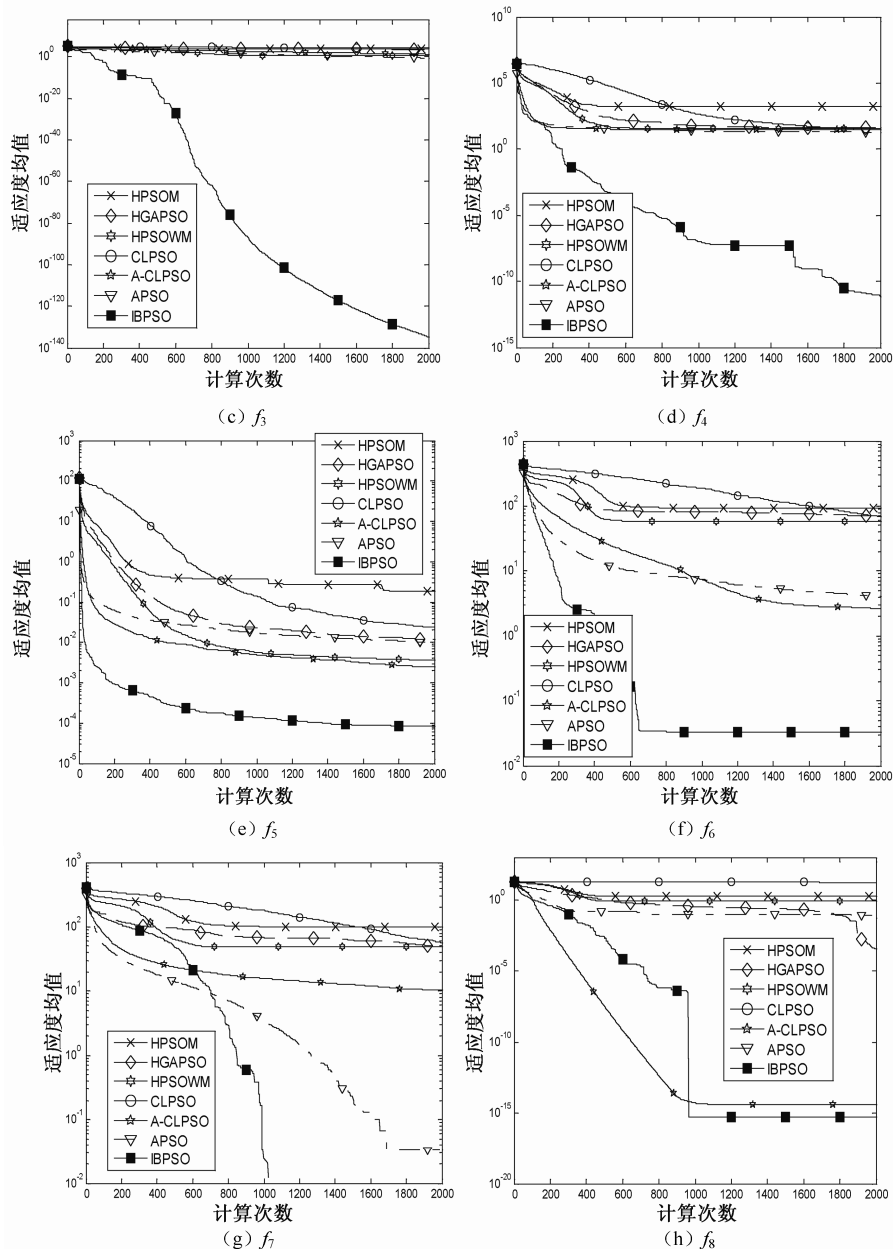


图 3.4 7 种变种粒子群算法收敛曲线比较 (续)

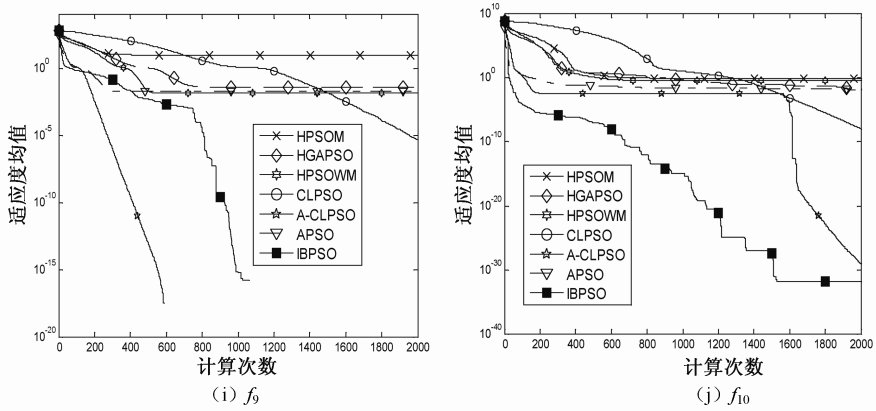


图 3.4 7 种变种粒子群算法收敛曲线比较 (续)

3.5.3 t-test 测试比较

$t\text{-test}^{[139]}$ 是一种用于区分两种算法优劣的统计方法, 如果 t 值为正, 则表示算法 1 的寻优性能优于算法 2, 反之, 则算法 2 的优化性能优于算法 1。

$$t = \frac{\bar{\alpha}_2 - \bar{\alpha}_1}{\sqrt{\left(\frac{\sigma_2^2}{\varepsilon + 1}\right) + \left(\frac{\sigma_1^2}{\varepsilon + 1}\right)}} \quad (3.21)$$

式 (3.21) 中 $\bar{\alpha}_1$ 、 $\bar{\alpha}_2$ 分别表示算法 1 与算法 2 的平均适应度值, σ_1 、 σ_2 分别表示算法 1 与算法 2 的标准方差, ε 表示自由度, 文中的自由度 $\varepsilon=49$ 。从数理统计 t 分布表可知, $t>0$ 表明算法 1 要优于算法 2, $t=0$ 表明算法 1 性能等于算法 2 性能, $t=1.645$ 表明算法 1 性能优于算法 2, 性能具有 95% 的置信度, $t=2.06$ 表明算法 1 性能优于算法 2 性能, 具有 98% 的置信度。

从表 3.5 中可以看出, 在 (f_1-f_{10}) 问题上 IBPSO 与其他几种变种的 PSO 算法的大部分 $t\text{-test}$ 值高于 1.645, 表明两算法区分度具有 95% 的置信水平。

表 3.5 IBPSO 算法与其他多种 PSOs 算法 t-test 比较

PSOs Functions	HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO
f_1	2.8323	3.6681	2.8526	17.4243	3.2974	4.0293
f_2	8.1646	4.9079	3.8602	34.1841	33.7418	0.9552
f_3	12.0041	6.5671	6.7699	38.6476	0.7419	7.87263
f_4	3.4811	8.0868	10.4823	22.1008	13.6880	10.9547
f_5	1.3841	16.8519	20.8987	32.5192	23.9529	25.3532
f_6	20.2544	21.0523	22.5567	78.8726	11.8898	18.6147
f_7	30.2315	12.5745	21.1084	59.6760	38.4273	1.3118
f_8	3.9606	2.3592	8.0318	38.7522	-	1.3114
f_9	2.3605	6.8746	7.3451	2.3690	N/A	8.3189
f_{10}	5.5178	2.2810	3.6682	11.0519	1.3117	2.2523

3.5.4 高维函数实验

高维函数优化具广泛的应用背景,科学、工程、社会经济等诸多领域都涉及该类问题,以往传统方法及智能优化算法在面对某些大型复杂问题时,往往须遍历整个搜索空间,将会产生搜索的组合爆炸现象,无法在规定时间内完成对任务的搜索。许多工程优化问题,如复杂系统控制器优化与设计,其搜索空间较大,同时对计算速度、收敛性都有严格的要求,因此研究解决高维函数优化的高效算法已成为科学工作者的研究目标之一。为了验证本章所提算法对高维问题的优化能力,本节选取 f_1 、 f_2 、 f_5 、 f_8 、 f_{10} 这 5 个典型优化函数,其测试维数为 200、500、800、1000、2000、3000 维,测试指标为优化值到达接受阈值内的平均计算次数、CPU 收敛时间以及收敛率。高维函数及其接受阈值设置如表 3.6 所示。算法达到接受阈值内自动退出此次迭代,算法最大迭代次数为 4000 代,算法测试 30 次取平均值。

表 3.6 测试高维函数及其接受阈值

测试高维函数	f_1	f_2	f_5	f_8	f_{10}
接受阈值 V_{accept}	1e-3	1e-5	1e-3	1e-3	1e-4

在这里, 平均计算次数 (Mean FEs), 计算机平均 CPU 时间 (Time), 平均收敛率 (Ratio) 等指标是在满足式 (3.22) 下完成的。

$$|F^* - F_{\text{best}}| \leq V_{\text{accept}} \quad (3.22)$$

上式中 F^* 为全局最优值, F_{best} 为此次运行所得的最优值, V_{accept} 为阈值, 如表 3.7 所示, 设 N 次运行中有 N_s 次成功, 则平均成功率为 $\text{Ratio}=N_s/N$ 。

表 3.7 高维函数测试结果 (30 次测试)

D (维数)		200	500	800	1000	2000	3000
f_1	Mean FEs	161.53	110.90	165.17	225.77	245.43	241.47
	Time(s)	5.3	12.96	40.49	80.57	288.51	582.97
	Ratio(%)	100	100	100	100	100	100
f_2	Mean FEs	690.77	776.23	771.80	819.50	712.63	634.63
	Time(s)	24.15	94.52	195.82	299.62	836.25	1538.55
	Ratio(%)	100	100	100	100	100	100
f_3	Mean FEs	101.33	90.03	89.20	108.90	127.03	85.67
	Time(s)	3.45	11.15	23.16	40.58	155.29	205.05
	Ratio(%)	100	100	100	100	100	100
f_8	Mean FEs	127	103.07	178	134.63	113.13	100.21
	Time(s)	5.08	13.97	48.98	52.89	139.98	249.81
	Ratio(%)	100	100	100	100	100	100
f_{10}	Mean FEs	123.80	161.06	125.57	108.50	123.30	129.63
	Time(s)	7.089	28.74	43.48	51.14	176.03	362.59

从表 3.7 可以得出 5 个测试函数算法在高维中都能收敛到可接受解范围内, 同时其迭代次数并不大, 从图 3.5 还可以看出 IBPSO 的函数值平均计算次数并没有随着维数的增大而增加, 对超高维也能够在几乎相同代数下达到可接受解, 说明 IBPSO 在高维函数优化方面有明显的优势, 单位搜索效率较高, 因为双模态动态机制以及免疫系统的受体编辑机制扩大了粒子群解区域的搜索空间, 提高了群体的多样性分布, 免疫克隆选择进一步增强群体逃离局部极值能力。同时随着维数的增大其 CPU 计算时间也相应增大, 这是因为维数增大, 其计算复杂度增大, 其时间代价还是可以接受的。通过实验表明本章 IBPSO 对高维系统具有较强的搜索能力。

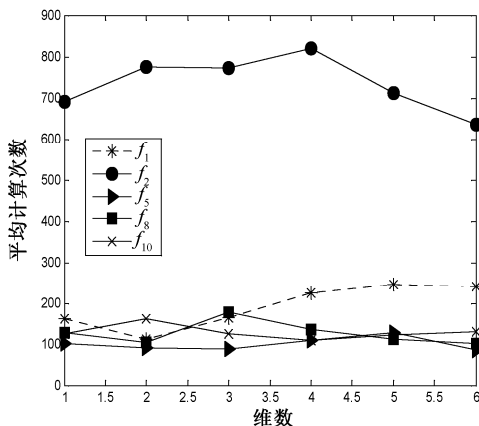


图 3.5 函数值计算次数与函数维数的关系曲线

3.5.5 多模态函数及多样性实验

在实际工程应用中,许多问题可转化为多模态优化问题,如复杂系统控制器设计、复杂控制器参数优化等。多模态函数的主要特点是存在多个全局最优解或若干个局部最优解,下面列举两个典型的多模态函数用于 IBPSO 算法多模态问题优化性能测试。

$$f_{11} = 1 + x \cdot \sin(4\pi x) - y \cdot \sin(4\pi y + \pi) + \frac{\sin(6\sqrt{x^2 + y^2})}{6\sqrt{x^2 + y^2 + 10^{-15}}}, \quad -1 \leq x, y \leq 1 \quad (3.23)$$

f_{11} 函数有 4 个全局最大值 2.118, 对称分布于 $(\pm 0.64, \pm 0.64)$ 。存在大量局部极大值,尤其是在中间区域有一取值与局部最大值接近的局部极大值 (2.007) 凸台。

$$f_{12}(x, y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2, \quad -6 \leq x, y \leq 6 \quad (3.24)$$

f_{12} 函数是改进的 Himmelbau 函数^[157], 为典型不可分、等高、非等距的二维多峰函数,在定义域内有 4 个相等的峰值,理论值均为 200.0 性能不佳的算法,很难搜索到全部 4 个峰值。

1. 免疫克隆选择算法对双态粒子群的影响

表 3.8 中 Best、Worst、Mean、Std·Dev、Time 表示测试获得的最优值、最差值、平均值、标准均方差、求解时间。ration 为同时找到全局多个极值点的百分比。BPSO 为双态粒子群算法,从表中可以看出 f_{11} , f_{12} 函数优化中能够搜索多模态函数的所有全局最优解,且每个最优解精度都达到了理想值,表明 IBPSO 算法多峰寻优能力较强,其原因是针对多个个体极值进行克隆变异的克隆选择机制和受体编辑引入有利于多峰全局寻优。由于免疫克隆选择的引入,会增加算法的时间复杂度,所以 IBPSO 的求解时间大于纯粹的双模态粒子群 BPSO,从表 3.8 中可以看出,其求解精度提高了,实际算法增加的 CPU 开销时间并不大,开销代价是可以接受的,证明人工免疫系统对算法寻优精度具有很好的贡献。

表 3.8 改进免疫算法对双态粒子群的影响 (30 次测试)

Function	method	Best	Worst	Mean	Std·Dev	Ration(%)	Time(s)
f_{11}	BPSO	2.1187634205	2.1187634193	2.1187634197	4.3e-10	90	3.8
	IBPSO	2.1187634205	2.1187634195	2.1187634198	4.5e-11	100	4.1
f_{12}	BPSO	200.0000	199.99999	199.99999	4.15e-14	68.33	4.0
	IBPSO	200.0000	200.00000	200.00000	0	100	6.0

Mean(x,y)为平均坐标,从表 3.9 中可以看出 IBPSO 能够搜索多峰函数的所有全局最优解,且每个最优解精度都达到了理想值,IBPSO 求解精度优于 NAFSA^[157] (生境人工鱼群算法),证明算法的全局寻优能力较强。

表 3.9 IBPSO 与 NAFSA 算法的比较 (f_{12})

Method	Mean(x,y)	Mean	Method	Mean(x,y)	Mean
IBPSO	(3.5844283310,-1.8481265036)	200.0000	NAFSA^[157]	(3.5861,-1.8479)	199.9999
	(3.0000000116,2.0000000021)	200.0000		(2.9989,1.9972)	199.9999
	(-2.8051180703,3.1313125328)	200.0000		(-2.8056,3.1331)	199.9998
	(-3.7793102444,-3.2831859960)	200.0000		(-3.7769,-3.2818)	199.9997

2. 多样性分析

多样性是评价粒子群算法的一种重要指标,多样性好坏关系到算法搜索是否趋于停滞。

多样性定义如下:

$$\text{diversity}(S) = \frac{1}{|L| \cdot |S|} \sum_{i=1}^{|S|} \sqrt{\sum_{j=1}^D (X_{ij} - \bar{X}_j)^2} \quad (3.25)$$

其中, S 为种群, $|S|$ 为种群所含粒子的个数, $|L|$ 为搜索空间的最长半径, D 为问题的维数, X_{ij} 为第 i 个粒子的第 j 个分量, \bar{X}_j 是全体粒子第 j 维的平均值。

由图 3.6 可看出基本 PSO 算法的种群多样性随着进化代数的增加而逐渐减少,多样性在进化后期快速降低。而 IBPSO 算法在进化过程中使种群的多样性得到了很好的保持,表明了 IBPSO 算法对改善种群多样性的有效性。双模态动态机制扩大了粒子群体解区间的搜索范围,平衡了 PSO 算法的开采和勘探能力,有效抑制了收敛过程中的早熟停滞现象,另一方面免疫系统的受体编辑机制提高了群体的多样性分布,免疫克隆选择进一步增强了群体逃离局部极值的能力。

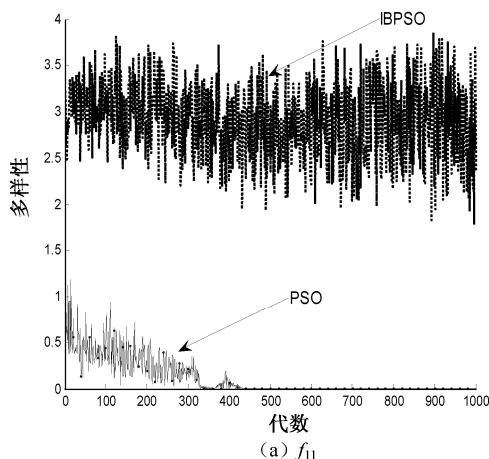


图 3.6 函数多样性变化曲线比较

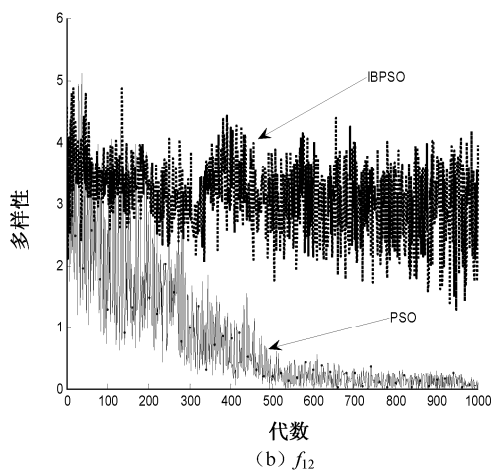


图 3.6 函数多样性变化曲线比较 (续)

3.6 本章小结

本章分析了基本粒子群算法易陷入局部极值的原因，结合人工免疫系统、鸟类及鱼群等微粒捕食行的特点，研究了一种免疫双态粒子群算法。一方面把粒子群划分为“捕食”与“探索”两种模态，扩大了解的搜索空间，平衡了 PSO 算法的开采和勘探能力，有效抑制了收敛过程中的早熟停滞现象，另一方面免疫系统克隆选择算法增强了群体逃离局部极值的能力，受体编辑机制提高了群体的动态活性。用经典测试函数进行仿真实验并将结果与多种改进的变种 PSO 进行比较，表明 IBPSO 算法在求解多模态问题及高维优化问题上优势明显，同时该算法收敛速度较快，实时性较好，比较适合复杂系统控制器的优化与设计。

第4章

免疫协同粒子群算法

在前两章分别研究了协同免疫进化算法及免疫双态粒子群算法，结合人工免疫系统、协同进化思想及粒子群的邻域信息的优势，本章将继续研究免疫协同粒子群进化算法，将提出一种免疫协同粒子群进化模型，扩大算法对问题解空间的搜索范围，提高算法对复杂问题的优化能力，分析相关参数、算子对算法的影响，最后对算法进行实验测试。

4.1 引言

随着粒子群算法的深入发展，人们开始将目光投向多群体粒子群算法研究。一些学者开始通过构造新的微粒组织或群结构来提高粒子群算法的收敛性能，于是多种群协同粒子群进化算法的研究开始逐渐得到报道^[158]。如 Bergh 等在文献[159]提出了一种多种群粒子群算法去优化不同维的求解向量。文献[160]研究了一种动态多群体粒子群进化算法，并设计了一种自适应微粒个数调节策略。Niu 等人在文献[161]提出了一种主从式结构的多群体粒子群优化方法，但该算法并未考虑如何避免微粒陷入局部极值的缺陷。同理，文献[162]中采用上下两层结构模型设计了一种双层多种群粒子群优化算法，底层各个基础种群相互独立进化，并从高层种群中得到优良信息指来导自身的进化；该文算法的双层进化机制增加了群体的随机性和多样性，提高了算法的全局搜索能力和收敛精度。文献[163]将物种概念引入粒子群算法，提出了一种基于物种概念的动态多种群粒子群优化算法，在进化过程中动态确定

物种,利用种群多样性信息动态调整物种半径,运用物种机理对解空间的不同区域进行搜索,最后将算法用于 SVM 的参数优化,并应用于电力系统短期负荷预测。文献[164]应用协同进化原理,通过混合策略选取各子系统的代表作为子系统间的信息交互,构造了一种分布式协同粒子群优化算法应用于解决电力系统无功优化问题,通过大规模电力系统无功优化问题分解成若干个子系统进行优化,从而有效地降低了求解问题的复杂度以提高全局寻优的能力,避免过早地陷入局部最优解。文献[165]提出一种基于单纯形法的多群体粒子群优化算法,算法采用多个种群,通过在奇数种群和偶数种群上分别并行运行粒子群算法与单纯形搜索方法,经过粒子群算法和单纯形法的协同搜索,提高了算法的收敛速度与求解精度。从上述可以看出,多群体粒子群算法提高了 PSO 算法寻优性能,现有多种群粒子群算法并未提供粒子群逃离局部极值机制,同时群体之间信息协同交互不够,算法出现了冗余迭代现象。

针对这一问题,借鉴协同进化思想可以扩大粒子群算法搜索空间,促使粒子多方向性广泛搜索,人工免疫系统具有全局优化能力,能使粒子群逃离局部极值点,并结合粒子群的邻域信息可以有效降低算法的冗余迭代,能有效提高算法对复杂问题的优化能力。将协同进化思想^[71]与免疫系统优化原理^[10]融入粒子群算法中,本章研究了一种免疫协同粒子群进化算法,算法采用并行协同计算框架,由记忆种群与若干个普通种群构成;在进化过程中普通种群中优秀个体进入记忆库种群。普通种群内部通过精英粒子保留、免疫网络以及柯西变异等混合策略共同产生新个体以保证群体的多样性;个体极值采用小波学习提高了微粒个体极值收敛性能;免疫克隆选择算法对记忆库进行精细搜索,迁移机制实现了整个种群信息共享与协同进化。实验结果表明该算法收敛精度高且全局搜索能力强,且具有较好的动态优化性能。

4.2 免疫协同粒子群进化算法

4.2.1 免疫协同粒子群进化模型

免疫协同粒子群进化算法是对粒子群算法的一种改进。它将协同进化思

想^[71]与免疫系统优化原理^[10]融入粒子群算法中。多群体协同进化扩大粒子群算法对解空间搜索范围,能有效提高算法对复杂问题的优化能力;人工免疫系统具有全局优化能力,能使粒子群逃离局部极值点,信息交互机制促进信息共享可以有效降低算法的冗余迭代。算法采用并行计算框架,能大幅度提高算法的收敛速度,同时能有效避免标准优化方法容易陷入局部最小值的问题。免疫协同粒子群进化算法系统模型如图4.1所示。

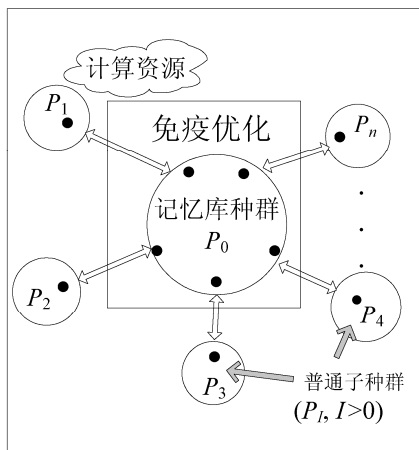


图4.1 免疫协同粒子群进化算法系统模型

图4.1表明, $P_0=(P_{11}, P_{12}, \dots, P_{ij})$ 为优良记忆种群, P_{ij} 代表第 i 个子种群的第 j 个代表。优良记忆种群 P_0 保存了各普通种群在进化过程中产生的优良模式,同时优良记忆库种群通过免疫克隆选择算法优化进行精细搜索,优势个体得到快速成长。通过迁移将全局最优的个体返回到各普通子种群中指导普通种群继续进化,加速了群体的收敛速度。 $P=(P_1, P_2, \dots, P_n)$ 为普通微粒种群,其中 P_i 代表第 i 个子种群。为了避免以往粒子群单一进化模式的缺点,提高普通粒子群的多样性能力及其全局寻优能力,普通种群内部通过精英粒子保留、免疫网络及柯西变异等混合策略产生新一代群体。普通种群中部分优秀个体极值采用自适应小波学习增强群体中个体极值进化速度。免疫协同粒子群进化算法具体流程如图4.2所示。

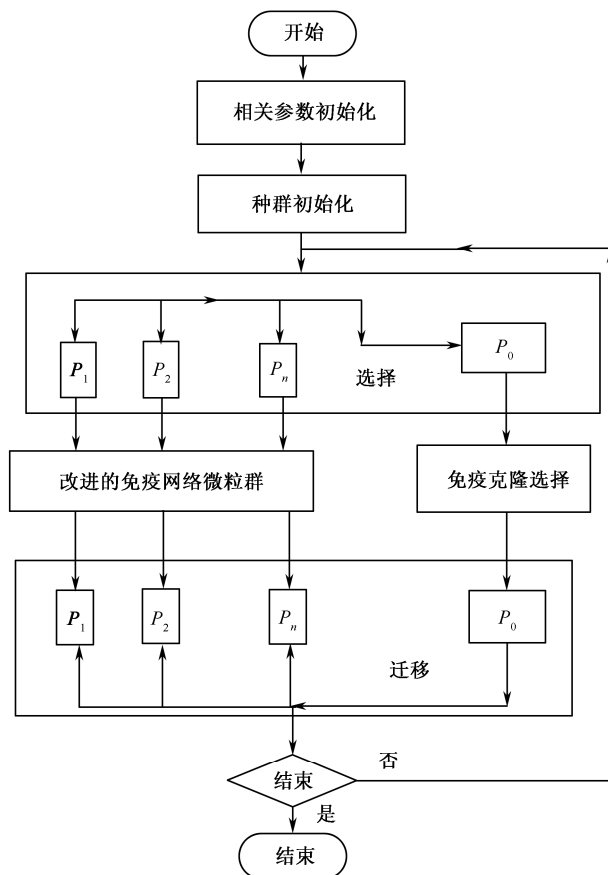


图 4.2 免疫协同粒子群进化算法并行协同框架

4.2.2 普通种群混合免疫网络粒子群算法

在第 1 章绪论中已经介绍了免疫网络理论学说，免疫网络学说^[32]表明：抗原-抗体、抗体-抗体之间形成了一种相互作用的动态网络体系结构，保持了机体适当的免疫强度和内部环境的稳定，同时还维持了抗体群的多样性。aiNet 新生成的网络细胞群 C^* 如下。

$$C^* = C - \alpha \cdot (C - C_{Ag}) \quad (4.1)$$

式中, C 代表网络细胞群, C_{Ag} 表示抗原细胞。 α 为网络突变率, 变异使抗体朝向识别抗原的方向进化。 C_{Ag} 与 α 的规模与 C 一样。式 (4.1) 中同时表明每个独立的抗体总是向抗体群中优秀的抗体学习来提高算法的收敛性能的。

受上述免疫网络理论启发, 在各普通子群体内部, 设计一种混合免疫网络粒子群算法模型, 通过精英粒子保留、免疫网络及柯西变异等策略共同产生下一代群体。将种群中的所有个体按适应度大小排序, 将其分成三部分, 通过下面三种混合策略共同作用产生算法新群体, 克服单一粒子群进化模式的缺陷。

① 精英个体保留策略, 将适应度值最优的前 N 个精英个体直接保留复制到下一代, 以保存进化过程中的最优个体。

② 免疫网络及柯西变异混合策略, 次优的 M 个粒子与精英粒子随机选择优秀个体, 通过免疫网络与柯西变异混合产生新 M 个子代个体。其子代个体生成方式如下式所示。

$$P_i(t+1) = P_i(t) - r_1 \cdot (P_i(t) - P_j(t)) + r_2 \cdot \text{Cauchy} \cdot P_i(t) \quad (4.2)$$

式中 $i \neq j$, 其中 P_j 为精英粒子中随机选取的粒子, r_1 、 r_2 均为 (0,1) 之间的随机数, 混合产生的个体不仅继承了父代的变异信息, 而且还注入了群体中优秀抗体基因。相比于传统的变异算子, Cauchy 算子具有较大的变异尺度, 其中 Cauchy 密度函数如下所示:

$$\text{Cauchy}_t(x) = \frac{1}{\pi} \frac{t}{t^2 + x^2}, -\infty < x < \infty, t > 0 \quad (4.3)$$

③ 重新初始化策略, 对适应度不高的 S 个粒子进行重新初始化, 有利于为群体补充新抗体。混合免疫网络粒子群算法结构如图 4.3 所示。

从图 4.3 可以看出, 该混合模式可以克服经典的粒子群搜索算法自身固有模式的缺陷, 提高了整体算法的搜索能力。

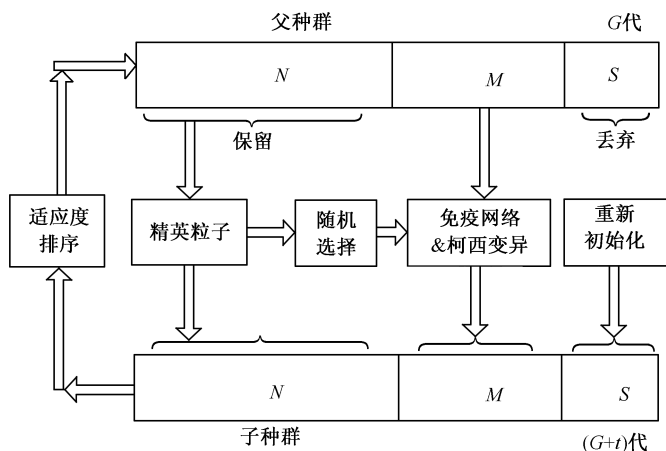


图 4.3 混合免疫网络粒子群算法

4.2.3 记忆库免疫进化

在每个子种群选择一个最优的个体（亲和度最高的个体）和随机选择一次优个体组成优势代表记忆库种群 $P_0^*=(P_{11},P_{12},\cdots,P_{ij})$ 。记忆库种群每一粒子都视为抗体，对免疫记忆库进行免疫优化操作，产生更优个体，可以加快优势抗体收敛到最优解，从而推动种群进化。

步骤 1：对记忆库种群 $P_0^*=(P_{11},P_{12},\cdots,P_{ij})$ 进行克隆扩增，克隆规模与亲和度成正比，克隆倍数 N_c 如下式所示：

$$N_c = \sum_i^N \text{round}(\frac{\beta N}{i} + b) \quad (4.4)$$

其中， N 为种群规模， $\beta \in (0,1)$ 。为了保证每个抗体都有一定克隆数量，故加上了常量 $b \geq 1$ 的整数。经过克隆扩增生成临时抗体群 Sub。

步骤 2：对临时抗体群 Sub 中的部分个体实施高频变异，其方法为基于变尺度的邻域内变异。其变异算子如下：

$$P_{id}^{\text{new}} = P_{id} + [(rd > P_m)] * \eta * P_{id} * U(0,1) - [(rd \leq P_m)] * \eta * P_{id} * U(0,1) \quad (4.5)$$

$$[rd > P_m] = \begin{cases} 1, & rd > P_m \\ 0, & \text{其他} \end{cases} \quad (4.6)$$

式中, rd 为随机数, $P_m=0.5$ 。

$$\eta(t) = 1 - r^{[1-(t/T)]^b} \quad (4.7)$$

式中, t 是群体当前进化代数, T 为总的进化代数, b 为正常数, 一般取值为 2, $r \in (0,1)$ 、 $U(0,1)$ 为 $(0,1)$ 之间的均匀随机数。从式 (4.7) 可看出, 在进化初期, 对于较小的 r 值, $\eta(t) \approx 1$, 此时的变异空间大; 而在进化后期, t 接近 T 时, $\eta(t) \approx 0$, 在小范围空间内进行局部搜索。

步骤 3: 免疫选择操作, 从克隆变异后的个体中选择亲和度最优的个体进入下一代。记忆库免疫优化流程如图 4.4 所示。

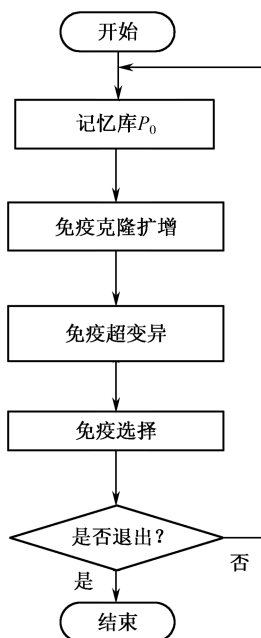


图 4.4 记忆库免疫优化流程

4.3 协作操作

粒子群中微粒个体极值在进化过程中有着重要的作用，群体中的全局最优值也产生于个体极值集合中，因此个体极值的寻优速度与求解精度影响到群体寻优效率，为了提高群体寻优性能，加强群体中个体极值信息交互，对群体微粒的个体极值进行自适应小波学习。

4.3.1 个体极值小波学习

1. 小波机理

定义 4.1: 设连续时间函数 $\psi(x)$ 为小波函数^[143]，满足以下性质。

$$\int_{-\infty}^{+\infty} \psi(x) dx = 0 \quad (4.8)$$

$$\int_{-\infty}^{+\infty} |\psi(x)|^2 dx < \infty \quad (4.9)$$

式 (4.8) 表明小波函数 $\psi(x)$ 的正总动量等于的负的总动量；式 (4.9) 表明小波函数 $\psi(x)$ 的幅值是有界限的，即大多能量都限制在有限的时空区域内，其幅值空间是动态变化的。小波函数动态变空间学习算子能克服以往固定半径学习算子的缺陷，该算子可以增强算法的动态优化性能。其母波函数如下所示。

$$\psi(x) = e^{-x^2/2} \cos(5x) \quad (4.10)$$

此处选择 Morlet 小波 (图 4.5)，如下式所示。

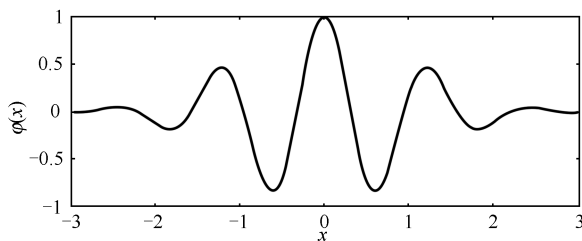


图 4.5 Morlet 小波图像

$$\sigma = \frac{1}{\sqrt{a}} e^{-\left(\frac{\varphi}{a}\right)^2/2} \cos\left[5\left(\frac{\varphi}{a}\right)\right] \quad (4.11)$$

其中 $\varphi \in [-2.5a, 2.5a]$ 。

2. 个体极值自适应小波

由图 4.6 可以得出随着参数 a 的增加小波幅值相应不断地减少, 为了避免手动调节 a 的烦琐, 于是提出自适应 a , 其表达式如下式所示。

$$a = a_{\min} + (a_{\max} - a_{\min}) \frac{t}{T} \quad (4.12)$$

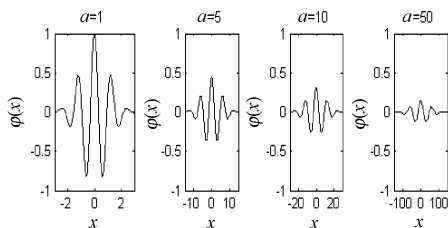


图 4.6 不同的参数 a 对小波幅值的影响

a_{\max} 与 a_{\min} 分别是 a 的上限与下限; 此时 a 随着代数自适应增加, 幅值 $\psi(x)$ 自适应动态微调。自适应小波图形如图 4.7 所示。

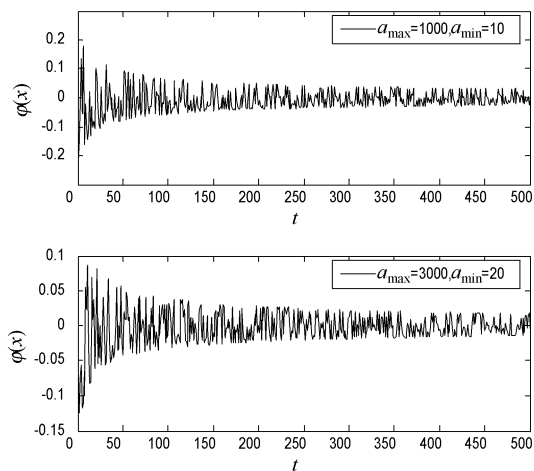


图 4.7 自适应小波图形

个体极值自适应小波学习策略是利用群体中历史最优信息来更新群体中个体极值微粒的位置，保证了粒子的多样性，同时加快了群体收敛速度。在免疫协同粒子群算法中，个体极值位置依据概率一部分向本群全局最好 Gbest 学习，另一部分向其他子种群 Gbest 学习。这样可以促进各个子群体间最优信息的交流，同时可以增强群体多样性抑制局部最优。

在免疫协同粒子群算法中个体极值自适应小波学习策略是利用群体中历史最优信息来更新群体中个体粒子的位置，保证了粒子的多样性，增强了算法动态优化性能，同时加快群体收敛速度。小波学习策略如下。

$$\text{Pbest}_j^d = \text{Pbest}_j^d + \sigma \cdot (\text{Gbest}_{P_i}^d - \text{Pbest}_j^d) \quad (4.13)$$

式中 σ 为自适应小波， Pbest_j^d 为普通种群中的个体极值， $\text{Gbest}_{P_i}^d$ 表示第 i 个普通种群的 Gbest 值，如果生成的随机数 rand 大于概率 p_c 则 P_i 选取的是自身群体的 Gbest 位置，否则是选取其他群体的 Gbest 位置。 P_i 的选取如图 4.8 所示。

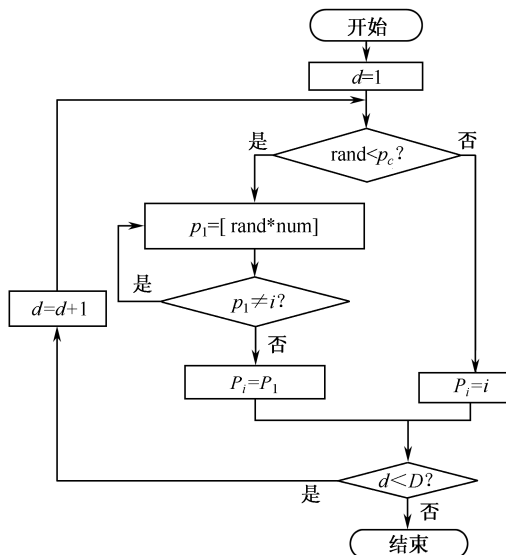


图 4.8 P_i 的选取流程图

4.3.2 优势抗体迁移与共享

优势抗体迁移：普通子种群内部进化过程中，每隔数代直接从优良种群库中引进若干优良种子个体（一般取一两个抗体）代替本种群中适应度较低的个体。操作如下：

其迁移信号 sig 为

$$\text{sig} = \begin{cases} 1, & \text{Acc}_i < 1 \\ 0, & \text{其他} \end{cases} \quad (4.14)$$

其中第 i 个种群的群体进化加速度 Acc_i 定义如下。

$$\text{Acc}_i = \frac{F_{\text{best}}^i(k) - F_{\text{best}}^i(k-t)}{F_{\text{best}}^i(k-t) - F_{\text{best}}^i(k-t-1) + \varepsilon} \quad (4.15)$$

上式中 $F_{\text{best}}^i(k)$ 、 $F_{\text{best}}^i(k-t)$ 分别表示第 $i(i=1, 2, \dots, n)$ 个普通种群当前和上 t 代进化过程中的最优适应度值。其中 ε 为平滑系数。上述表明，种群 i 在进化的过程中连续出现减慢现象，需要从优良种群库中迁移优良抗体来提高该子种群的优化性能。普通子种群内部进化过程中，每隔数代当迁移信号 $\text{sig}=1$ 时直接从优良种群库中引进若干优良种子个体（一般取一两个抗体）代替本种群中适应度较劣且成长性不佳的个体。通过迁移操作可以将记忆库中通过免疫优化提升操作的优秀抗体引入普通种群中，实现了种群信息交流与共享，加速推进了群体进化收敛性能。

4.4 ICPSO 算法流程

ICPSO 算法流程如图 4.9 所示。

4.5 实验仿真及分析

为了验证文中提出的免疫协同粒子群进化算法（ICPSO）的收敛速度、

全局搜索性能。在这里选用 10 个典型的复杂函数进行测试。表 4.1 中, $f_1 \sim f_5$ 是单峰函数, 通常用于检验算法收敛速度; $f_6 \sim f_{10}$ 是非线性多峰函数, 存在大量的局部极值点, 可用于检验算法的多样性、全局优化性能、逃离局部极值能力与避免早熟收敛能力。

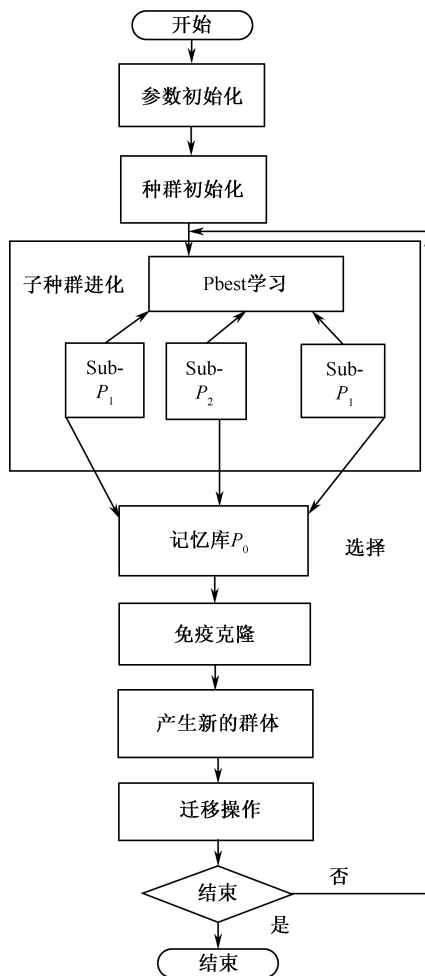


图 4.9 ICPSO 算法流程

实验设计: 实验条件为 AMD Athlon(tm) II X2 250 处理器, 主频 3.0GHz, 3.25GB 内存, Windows XP2, Visual C++ 6.0 平台。为了减少实验统计误差, 所有实例独立运行 30 次, 最大迭代次数为 3000 代, 对平均统计结果进行比较。文中 HPSOM 算法表示基于随机均匀变异的粒子群算法^[156], HGAPSO 为基于遗传算法的混合粒子群算法^[141], HPSOWM 为基于小波变异的粒子群算法^[143], CLPSO 为全面学习粒子群算法^[147], A-CLPSO 为自适应全面学习粒子群算法^[148], APSO 表示一种自适应模糊分类粒子群算法^[146]。实验参数设置: HPSOM、HGAPSO、HPSOWM 参数设置参照文献[143]。CLPSO 算法中 $\omega \in [0.9, 0.4]$, $c_1=c_2=1.49445$; APSO 算法中 $\omega \in [0.9, 0.4]$, $c_1=c_2=2.0$ 。为了保证测试公平, 所有粒子群算法单个种群粒子规模为 50, 文中 ICPSO 算法 $\omega \in [0.91, 0.4]$, $c_1=c_2=1.49445$ 。ICPSO 种群规模为 4 较合适, 种群数目的设定在下述实验的分析中得到。

表 4.1 测试函数

测试函数	作用域	V_{accept}	全局最优
$f_1(x) = \sum_{i=1}^{30} x_i^2$	$-100 \leq x_i \leq 100$	0.001	0
$f_2(x) = \sum_{i=1}^{30} x_i + \prod_{i=1}^{30} x_i $	$-10 \leq x_i \leq 10$	0.00001	0
$f_3(x) = \sum_{i=1}^{30} (\sum_{j=1}^i x_j)^2$	$-100 \leq x_j \leq 100$	0.001	0
$f_4(x) = \sum_{i=1}^{30} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$-10 \leq x_i \leq 10$	50	0
$f_5(x) = \sum_{i=1}^{30} ix_i^4 + \text{random}(0, 1)$	$-1.28 \leq x_i \leq 1.28$	0.01	0
$f_6(x) = \sum_{i=1}^{30} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$-500 \leq x_i \leq 500$	0.01	0
$f_7(x) = \sum_{i=1}^{30} [y_i^2 - 10 \cos(2\pi y_i) + 10]$ where, $y_i = \begin{cases} x_i, & x_i < 0.5 \\ \text{round}(2x_i)/2, & x_i \geq 0.5 \end{cases}$	$-5.12 \leq x_i \leq 5.12$	0.001	0

续表

测试函数	作用域	V_{accept}	全局最优
$f_8(x) = -20 \exp(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i}) - \exp[\frac{1}{30} \sum_{i=1}^{30} \cos(2\pi x_i)] + 20 + e$	$32 \leq x_i \leq 32$	0.001	0
$f_9(x) = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$-600 \leq x_i \leq 600$	0.001	0
$f_{10}(x) = \frac{\pi}{30} \{10 \sin(\pi y_1) + \sum_{i=1}^{30-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_{30} - 1)^2\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4)$ $\text{where, } y_i = 1 + \frac{1}{4} (x_i + 1), u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$-50 \leq x_i \leq 50$	0.0001	0

4.5.1 算法精度比较

求解精度是衡量算法的一个重要指标, 表 4.2 中表述了算法收敛精度与标准方差, 图 4.10 表述了 10 个测试测试函数多种 PSO 算法的收敛性能。表 4.2 中 Mean、Std.Dev 分别表示适应度的均值、标准方差。从表 4.2 中可以看出在 f_1 、 f_2 、 f_3 、 f_4 、 f_5 、 f_6 、 f_{10} 函数问题上, ICPSO 算法的均值与标准方差明显优于其他 6 种变种粒子群算法, 在 f_1 问题上 ICPSO 算法在均值与方差方面优于 HPSOM、HGAPSO、HPSOWM、CLPSO、A-CLPSO 算法; 同时在 f_4 问题上 ICPSO 算法均值及标准方差与 APSO 相等; 在 f_9 问题上, ICPSO 算法在均值与方差方面优于 HPSOM、HGAPSO、HPSOWM、CLPSO、APSO 算法, 同时在 f_4 问题上 ICPSO 算法均值及标准方差与 A-CLPSO 算法相等; 但从图 4.10 可以看出 ICPSO 算法的收敛速度比 A-CLPSO、APSO 算法快。从图 4.10 可以得出 ICPSO 算法收敛速度比其他 6 种变种 PSO 算法快。上述实验结果表明 ICPSO 算法在单模态问题与多模态问题上寻优能力都很强, 其算法收敛精度及算法稳健性较好。

表 4.2 七种变种粒子群算法精度比较 (30 次)

Function		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	ICPSO
f_1	Mean	9.05×10^{-12}	2.43×10^{-13}	8.50×10^{-12}	1.63×10^{-16}	2.58×10^{-139}	1.15×10^{-64}	1.02×10^{-284}
	Std.Dev	2.70×10^{-11}	1.10×10^{-12}	3.63×10^{-11}	8.33×10^{-17}	1.12×10^{-138}	4.48×10^{-64}	0
f_2	Mean	5.33	3.67	9.89×10^{-5}	3.86×10^{-10}	2.03×10^{-80}	3.66×10^{-41}	3.37×10^{-318}
	Std.Dev	7.18	7.07	2.62×10^{-4}	1.25×10^{-10}	9.47×10^{-80}	7.55×10^{-41}	0
f_3	Mean	10149.8	1866.66	0.27	10447.6	1.13×10^{-2}	4.15×10^{-3}	8.61×10^{-147}
	Std.Dev	6761.24	2874.01	0.31	1643.58	8.10×10^{-3}	3.23×10^{-3}	4.61×10^{-146}
f_4	Mean	2072.59	40.14	27.67	26.70	26.88	24.81	7.53×10^{-7}
	Std.Dev	3978.51	34.94	17.28	4.89	15.81	21.88	8.19×10^{-7}
f_5	Mean	0.188	8.65×10^{-3}	1.74×10^{-3}	1.48×10^{-2}	1.51×10^{-3}	6.63×10^{-3}	1.34×10^{-4}
	Std.Dev	0.67	2.93×10^{-3}	7.80×10^{-4}	2.46×10^{-3}	5.55×10^{-4}	2.46×10^{-3}	7.45×10^{-5}
f_6	Mean	106.63	57.01	50.44	57.71	5.57	2.09	0
	Std.Dev	32.66	19.95	14.03	7.08	2.13	1.37	0
f_7	Mean	93.57	42.30	51.6	45.34	7.72	0	0
	Std.Dev	19.65	22.63	21.41	6.67	2.40	0	0
f_8	Mean	2.63	3.61×10^{-8}	0.48	18.45	4.14×10^{-15}	1.07×10^{-14}	8.25×10^{-16}
	Std.Dev	4.89	4.06×10^{-8}	0.65	2.73	0	1.85×10^{-15}	8.86×10^{-16}
f_9	Mean	1.57×10^{-2}	3.48×10^{-2}	5.01×10^{-3}	8.73×10^{-8}	0	1.03×10^{-2}	0
	Std.Dev	1.97×10^{-2}	4.22×10^{-2}	9.33×10^{-3}	2.82×10^{-7}	0	9.38×10^{-3}	0
f_{10}	Mean	0.447	7.10×10^{-6}	0.187	3.56×10^{-15}	1.57×10^{-32}	1.04×10^{-2}	1.57×10^{-32}
	Std.Dev	0.693	2.92×10^{-5}	0.314	2.41×10^{-15}	2.74×10^{-48}	3.11×10^{-2}	2.74×10^{-48}

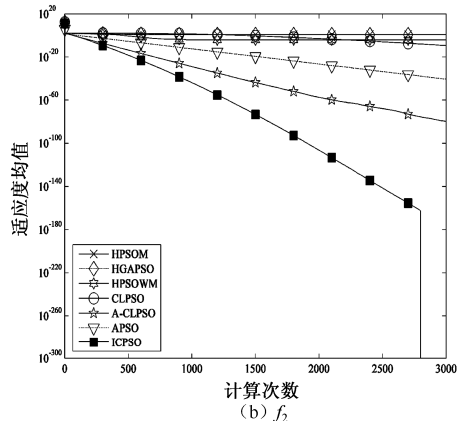
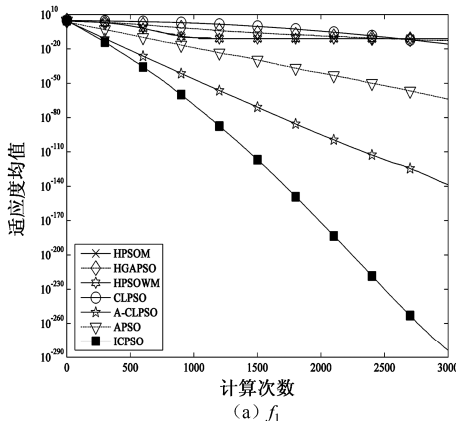


图 4.10 七种算法收敛性曲线

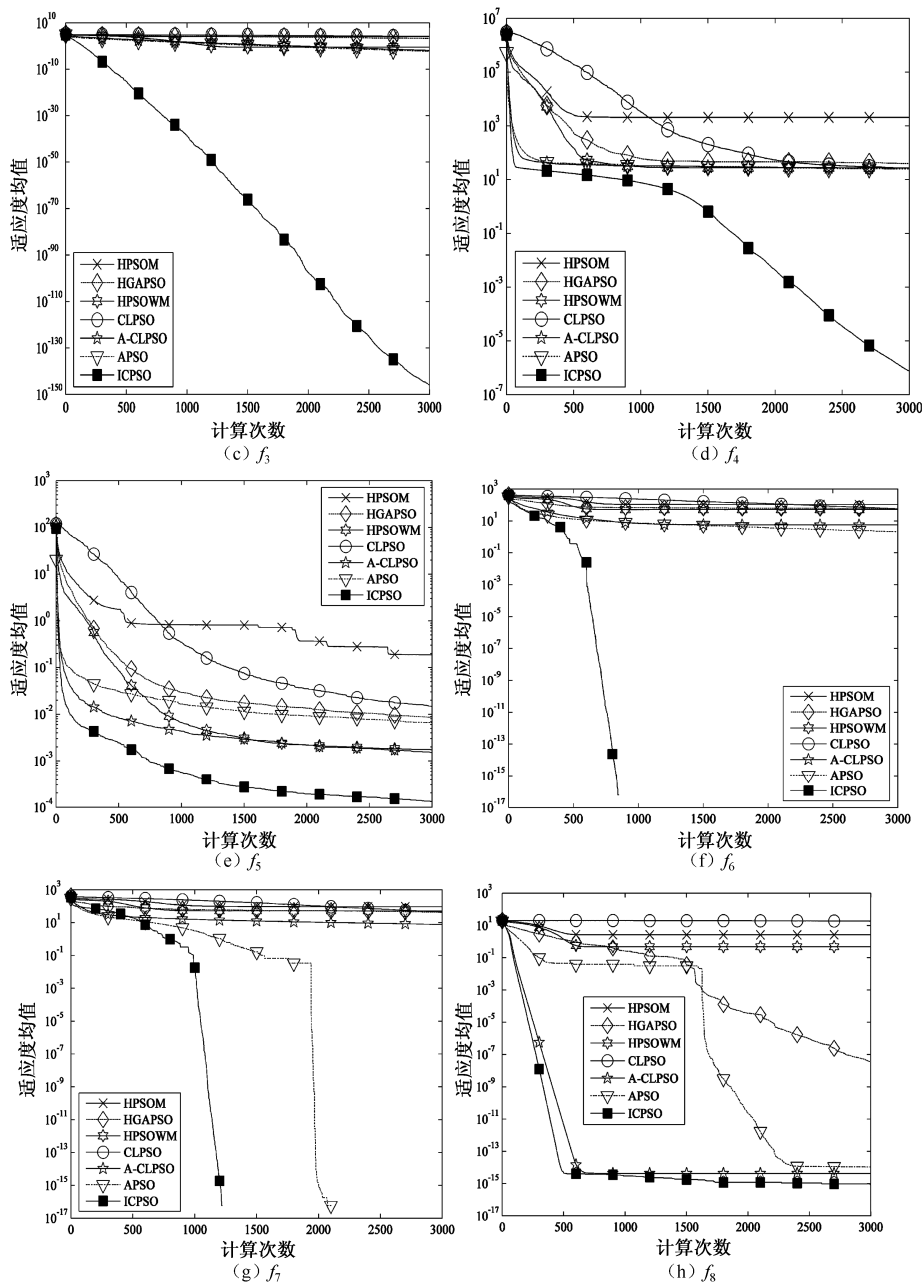


图 4.10 七种算法收敛性曲线 (续)

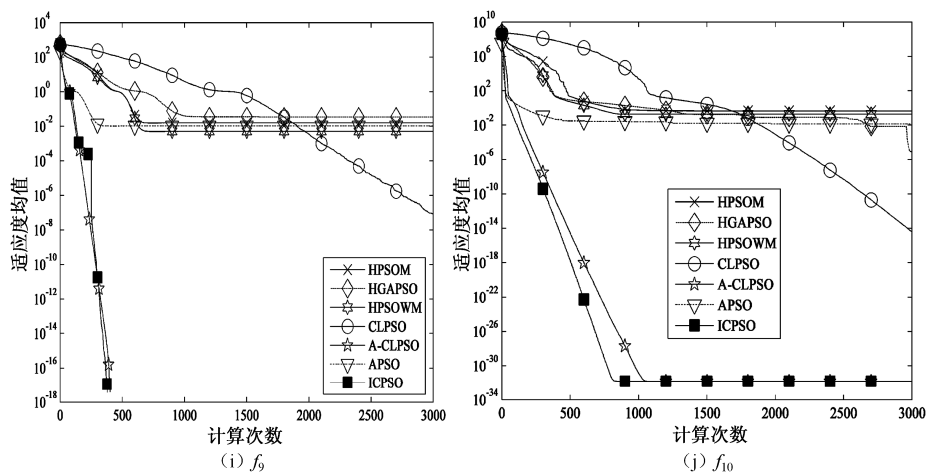


图 4.10 七种算法收敛性曲线 (续)

4.5.2 算法计算复杂度比较

平均计算次数 (Mean FEs) 和 CPU 平均计算时间 (Time) 是评价一个算法计算复杂度的重要指标。表 4.3 中给出了多种算法在达到可接受解阈值内 (V_{accept}) 的平均计算次数、CPU 实际计算平均时间及收敛率。

在本节, 平均计算次数、计算机平均 CPU 时间、平均收敛率等指标是在满足式 (4.16) 下完成的。

$$|F^* - F_{\text{best}}| \leq V_{\text{accept}} \quad (4.16)$$

上式中 F^* 为全局最优值, F_{best} 为此次运行所得的最优值, V_{accept} 为解的接受阈值, 如表 4.1 所示, 设 N 次运行中有 N_s 次成功, 则平均成功率为 $\text{Ratio} = N_s / N$ 。

表 4.3 中 “-” 表明在设定的条件内没能得到收敛。从表 4.3 可以看出, 在 10 个函数测试实例上, ICPSO 算法平均计算次数明显少于其他 6 种算法, 以函数 f_1 为例, 其他 6 种粒子群算法 HPSOM、HGAPSO、HPSOWM、CLPSO、A-CLPSO、APSO 分别平均需要 1096.5、2050.3、1149.0、3766.4、205.8、283.2 迭代次数, 而 ICPSO 平均需要代数仅为 89.87 次, 说明 ICPSO 单步迭代优化效率高。在单处理器上 ICPSO 模型具有两层结构, 内有双层嵌套循环, 理论上比文中所提其他算法 (仅双层嵌套循环) 复杂度高。但求解

精度达到可接受解阈值的实际 CPU 开销时间对算法计算复杂度更具有价值,从表 4.3 还可以看出在 10 个测试实例上,ICPSO 的 CPU 实际计算时间代价均少于其他 6 种变种粒子群算法,以具体函数 f_1 为例,其他几种变种粒子群算法 HPSOM、HGAPSO、HPSOWM、CLPSO、A-CLPSO、APSO 达到可接受阈值内平均需要 0.228s、0.406s、0.244s、0.520s、0.077s、0.139s、而本书 ICPSO 仅平均需要 0.056s,说明本书的算法在可接受解阈值内 CPU 时间代价相对较低,证明算法的求解效率较高。在上述 10 个标准测试函数问题上,在可接受阈值内,ICPSO 算法具有 100%的收敛成功率,说明 ICPSO 算法稳健性较好。ICPSO 算法具有的优点:多群体协同进化使粒子群算法对解空间搜索范围扩大,能有效提高算法对复杂问题的优化能力,人工免疫系统具有全局优化能力,能使粒子群逃离局部极值点,信息交互机制可以降低算法的冗余迭代。所以 ICPSO 算法平均计算次数、CPU 平均计算时间、收敛率都有明显的优势。

表 4.3 7 种算法计算复杂度比较 (30 次测试)

Function		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	ICPSO
f_1	Mean FEs	1096.5	2050.3	1149.0	3766.4	205.8	283.2	89.87
	Time(s)	0.228	0.406	0.244	0.520	0.077	0.139	0.056
	Ratio(%)	100.0	100.0	100.0	100.0	100.0	100.0	100.0
f_2	Mean FEs	1411.5	4579.2	1382.1	4489.1	316.4	459.9	184.40
	Time(s)	0.319	0.981	0.313	0.667	0.124	0.228	0.121
	Ratio(%)	60.0	63.3	96.7	100.0	100.0	100.0	100.0
f_3	Mean FEs	2738.0	6304.3	2623.3	—	5351.6	3248.5	240.13
	Time(s)	1.281	2.986	1.228	—	2.960	2.315	0.218
	Ratio(%)	33.3	60.0	100.0	0	100.0	100.0	100.0
f_4	Mean FEs	1411.8	2049.6	1253.7	4382.6	563.8	661.6	90.67
	Time(s)	0.324	0.458	0.285	0.635	0.231	0.334	0.131
	Ratio(%)	73.3	93.3	100.0	100.0	100.0	100.0	100.0
f_5	Mean FEs	2387.5	3559.2	1809.4	5535.4	386.80	1737.1	178.93
	Time(s)	0.521	0.657	0.399	0.822	0.134	0.852	0.116
	Ratio(%)	96.7	100.0	100.0	100.0	100.0	100.0	100.0
f_6	Mean FEs	—	1674.0	—	—	4015.2	4499.5	516.60
	Time(s)	—	0.484	—	—	1.691	2.531	0.889
	Ratio(%)	0	33.3	0	0	16.7	90.0	100.0

续表

Function		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	ICPSO
f_7	Mean FEs	—	3393.0	—	—	6705.7	1295.4	716.03
	Time(s)	—	1.227	—	—	3.047	0.805	0.652
	Ratio(%)	0	6.67	—	0	10.0	100.0	100.0
f_8	Mean FEs	1265.3	2572.1	1285.0	—	246	438.9	113.97
	Time(s)	0.470	0.726	0.472	—	0.110	0.250	0.085
	Ratio(%)	96.7	100.0	96.7	0	100.0	100.0	100.0
f_9	Mean FEs	1139.2	2148.6	1180.3	5164.6	212.0	301.3	81.93
	Time(s)	0.461	0.738	0.481	1.319	0.107	0.186	0.105
	Ratio(%)	56.7	30.0	56.7	100.0	100.0	33.3	100.0
f_{10}	Mean FEs	1721.4	2135.4	1507.0	4570.6	207.6	1129.7	227.70
	Time(s)	1.338	1.222	1.128	1.998	0.132	0.858	0.546
	Ratio(%)	56.7	100.0	66.7	100.0	100.0	100.0	100.0

4.5.3 算法 t-test 值比较

从表 4.4 中可以看出, 在 ($f_1 \sim f_{10}$) 问题上 ICPSO 与其他几种变种的 PSO 算法的 t-test 值均高于 1.645, 表明两算法区分度具有 95% 的置信水平。在 f_{10} 问题上 ICPSO 与 A-CLPSO 的 t-test 值等于 0, 表明 ICPSO 与 A-CLPSO 算法性能相当。

表 4.4 ICPSO 算法对其他变种粒子群算法的 t-test 值

PSOs Functions	HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO
f_1	2.3701	1.5621	1.6558	13.8365	1.6289	1.8151
f_2	5.2491	3.6706	2.6692	21.8355	1.5158	3.4278
f_3	10.6149	4.5926	6.1587	44.9480	9.8646	9.0851
f_4	3.6836	8.1234	11.3227	38.6089	12.0222	8.0180
f_5	1.9827	20.5453	14.4932	42.1369	17.3753	18.6637
f_6	23.0860	20.2066	25.4216	57.6372	18.4910	10.7872
f_7	33.6712	13.2172	17.0419	48.0663	22.7453	N/A
f_8	3.8030	6.2873	5.2217	47.7880	26.4566	34.0416
f_9	5.6353	5.8311	3.7970	2.1890	N/A	7.7646
f_{10}	4.5610	1.7193	4.2111	10.4452	0	2.3646

4.5.4 相关参数及其算子分析

接下来就种群数目、免疫克隆选择算子、自适应小波学习算子对算法解的质量影响进行实验分析与讨论。

从表 4.5 中可以看出,当 ICPSO 算法子种群数目为 3 时,其收敛精度与适应度均方差明显劣于子种群数目为 4;当子种群数目设为 4 的收敛精度与适应度均方差和子种群数 5 的收敛性能大体上相当。然而随着普通子种群数目的增加,算法的收敛性能也相应地增强,但算法时间复杂度也相应增大,同时考虑到收敛精度与平均方差的增强优势不是特别明显,综合权衡算法收敛精度与时间复杂度,设定子种群数目为 4 较合适。

表 4.5 种群数目对算法的影响

种群数	指标	f_1	f_3	f_4	f_5	f_8	f_{10}
3	Mean	4.26e^{-280}	1.26×10^{-132}	9.19×10^{-2}	2.11×10^{-4}	1.42×10^{-15}	1.57×10^{-32}
	Std.dev	0	6.78×10^{-132}	0.274	1.45×10^{-4}	1.50×10^{-15}	2.74×10^{-48}
	Time(s)	9.15	14.2547	9.203	10.352	10.890	17.087
4	Mean	1.02×10^{-284}	8.61×10^{-147}	7.53×10^{-7}	1.34×10^{-4}	8.25×10^{-16}	1.57×10^{-32}
	Std.dev	0	4.61×10^{-146}	8.19×10^{-7}	7.45×10^{-5}	8.86×10^{-16}	2.74×10^{-48}
	Time(s)	13.31	17.716	12.620	13.780	13.745	22.833
5	Mean	6.34e^{-286}	5.53×10^{-149}	1.04×10^{-7}	1.17×10^{-4}	9.44×10^{-16}	1.57×10^{-32}
	Std.dev	0	2.98×10^{-148}	1.08×10^{-7}	7.38×10^{-5}	1.07×10^{-15}	2.74×10^{-48}
	Time(s)	15.31	23.630	15.404	17.524	17.018	28.555
6	Mean	1.01×10^{-287}	7.36×10^{-153}	1.48e^{-8}	7.44×10^{-5}	1.18×10^{-15}	1.57×10^{-32}
	Std.dev	0	3.95×10^{-152}	1.71e^{-8}	6.29×10^{-5}	1.32×10^{-15}	2.73×10^{-48}
	Time(s)	18.32	25.89	18.8	20.66	21.31	35.06

从表 4.6 中可以看出,带有小波学习机制的 ICPSO 算法(ICPSO with Wavelet)的收敛精度优于没带有小波学习机制的 ICPSO 算法(ICPSO with out Wavelet)。小波学习机制能有效加快算法的收敛精度,算法在精度提高的同时,其 CPU 时间复杂度却相应增加。

表 4.6 免疫克隆选择算子及自适应小波学习对算法质量的影响

		f_1	f_3	f_4	f_5	f_8	f_{10}
ICPSO with Wavelet	Mean	4.39×10^{-275}	3.93×10^{-90}	18.18	1.77×10^{-4}	2.36×10^{-15}	1.57×10^{-32}
	Std.dev	0	1.49×10^{-89}	1.08	7.53×10^{-5}	1.99×10^{-15}	2.74×10^{-48}
ICPSO with AIS	Mean	5.45×10^{-284}	1.25×10^{-143}	2.51×10^{-6}	1.41×10^{-4}	2.96×10^{-15}	1.57×10^{-32}
	Std.dev	0	6.69×10^{-143}	4.43×10^{-6}	1.04×10^{-4}	1.67×10^{-15}	2.74×10^{-48}
ICPSO with AIS& Wavelet	Mean	1.02×10^{-284}	8.61×10^{-147}	7.53×10^{-7}	1.34×10^{-4}	8.25×10^{-16}	1.57×10^{-32}
	Std.dev	0	4.61×10^{-146}	8.19×10^{-7}	7.45×10^{-5}	8.86×10^{-16}	2.74×10^{-48}
ICPSO without Either	Mean	2.47×10^{-274}	2.36×10^{-69}	24.47	2.12×10^{-4}	0.75	1.94×10^{-17}
	Std.dev	1.05×10^{-282}	1.27×10^{-68}	22.88	1.02×10^{-4}	3.57	1.04×10^{-14}

表 4.6 中 (ICPSO with AIS) 表示 ICPSO 算法高层记忆库经过免疫克隆选择算法优化; (ICPSO without AIS) 高层记忆库未经免疫克隆选择算法优化。从表中可以得出, 免疫克隆选择算法对 ICPSO 算法影响较大, 从上述测试实例可以得出记忆库种群经过免疫克隆选择算子优化的求解精度明显高于未经过免疫克隆算子优化的 ICPSO 算法, 免疫克隆选择算子对记忆库进行了提升操作, 致使优良个体跳出了局部极值; 同时通过迁移操作, 优良信息在普通种群中传播, 实现了协同进化。带有免疫克隆选择算子的 ICPSO 算法的 CPU 消耗时间并未明显高于没带有免疫克隆选择算子的 ICPSO 算法, 带有免疫克隆选择算子的 ICPSO 算法的 CPU 消耗时间少于没带有免疫克隆选择算子的 ICPSO 算法。这说明免疫克隆选择算子协助优秀个体跳出局部极值, 有助于加快收敛速度, 从而避免了一些算子操作。因此免疫克隆选择算子并未增加整个算法的时间复杂度。

4.5.5 动态性能测试

为了避免标准测试函数取得全局最优点的参变量相同的情况, 选用 Shift 动态函数来测试算法的动态优化性能, Shift 函数^[166]取得全局最优点的参变

量具有不同的值,同时全局最优点也是随机生成的,具有移动不确定性,这样更能测试算法的全局动态寻优能力,通常用于动态测试。

Shift 函数构成方式为 $F(x) = f(x - o_{\text{new}} + o_{\text{old}})$, 其中 $F(x)$ 是新合成函数, $f(x)$ 是原函数, o_{old} 是原函数的全局最优点, o_{new} 是新合成函数的全局最优点。选用如下四个典型动态优化函数进行实验,其函数表达式如表 4.7 所示。

在 Shift 动态函数问题求解上,从表 4.8、图 4.11 中可以看出 ICPSO 算法在全局最优值、标准方差、t-value 值及收敛速度上都表现出很好的性能,表明文中所提算法 ICPSO 具有很好的动态寻优能力及抗干扰能力。相比其他算法而言,本书 ICPSO 算法具有较好的动态性能,主要由于免疫协同粒子群技术既能保持群体多样性又能较好地平衡全局与局部搜索能力,因此能较好地跟踪环境变化,同时还有很好的稳定性,适合于复杂非线性动态问题求解。

表 4.7 Shift 函数测试

Shift 测试函数	作用域	Z	全局最优
$F_1(x) = \sum_{i=1}^{30} z_i^2 - 450$	$-100 \leq x_i \leq 100$	$z = X - o$	-450
$F_2(x) = \max\{ z_i , 1 \leq i \leq D\} - 450$	$-100 \leq x_i \leq 100$	$z = X - o$	-450
$F_3(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos(\frac{z_i}{\sqrt{i}}) - 180$	$-600 \leq x_j \leq 600$	$z = X - o$	-180
$F_4(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i}) - \exp[\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)] + 20 + e - 140$	$-32 \leq x_i \leq 32$	$z = X - o$	-140

表 4.8 七种算法 Shift 函数测试结果比较

Function		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	ICPSO
f_1	Mean	6915.91	-449.94	-450	-450	-450	-450	-450
	Std.Dev	5669.53	0.165	6.89e-10	5.59e-10	0	1.80e-14	0
	t-value	9.1868	2.5713	0	0	N/A	0	N/A
f_2	Mean	-400.94	-446.89	-449.34	-446.59	-449.935	-449.925	-450
	Std.Dev	9.25	2.967	0.385	0.480	0.115	5.20e-02	2.02e-05
	t-value	37.5034	7.4119	12.1218	50.2340	3.9967	10.1987	0

续表

Function		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	ICPSO
f_3	Mean	-88.737	-179.70	-179.956	-179.839	-179.999	-179.989	-179.993
	Std.Dev	49.6824	0.247	8.61e-02	0.532	2.49e-03	1.02e-02	9.89e-03
	t-value	12.9880	8.3812	3.0188	2.0465	-4.1600	1.9908	0
f_4	Mean	-126.02	-132.58	-139.329	-140	-140	-139.778	-140
	Std.Dev	2.827	5.557	1.110	7.71e-08	0	0.449	0
	t-value	34.9676	9.4417	4.2745	0	N/A	3.4962	N/A

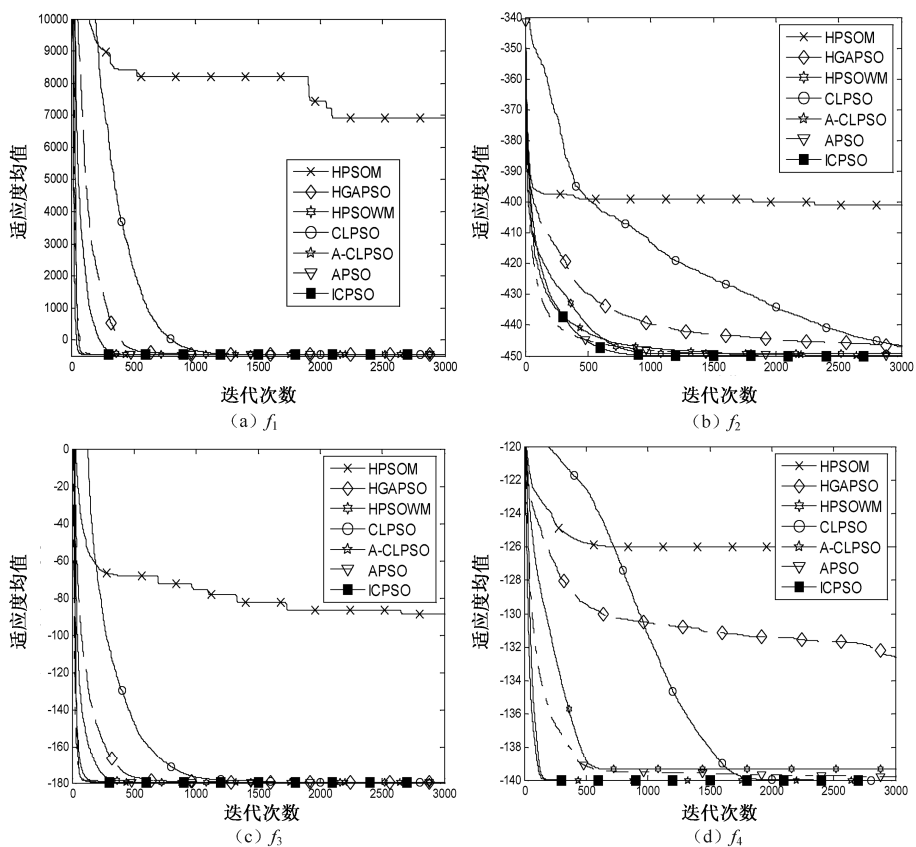


图 4.11 7 种算法 Shift 函数收敛曲线比较

4.5.6 ICPSO 与 IBPSO 的性能比较

为了验证 ICPSO 与 IBPSO 算法各自的优势,在第 3 章中研究的免疫双态粒子群算法 (IBPSO) 在多模态问题、高维优化问题求解方面具有优势,同时其算法实时性较好。

为了测试 IBPSO 与 ICPSO 在计算时间复杂度上的优势,在这里选用表 4.1 中 8 个典型优化函数进行实验,其设定条件如表 4.1 所示,实验结果如表 4.9 所示。表 4.9 中给出了多种算法在达到可接受解阈值内 (V_{accept}) 的平均计算次数、CPU 实际计算平均时间及收敛率。从表 4.9 实验中可以得出,8 个问题测试证明 IBPSO 算法的平均计算次数、CPU 平均计算时间都优于 ICPSO 算法,表明 IBPSO 算法在时间复杂度方面优于 ICPSO 算法,具有较好的实时性。

表 4.9 ICPSO 与 IBPSO 算法的计算复杂度比较 (30 次测试)

算法	指标	f_1	f_2	f_3	f_4	f_5	f_6	f_8	f_{10}
IBPSO	Mean FEs	69	178.25	75	43	20.34	109	83.67	101.7
	Time(s)	0.039	0.094	0.057	0.027	0.011	0.171	0.057	0.239
	Ratio(%)	100	100	100	100	100	100	100	100
ICPSO	Mean FEs	89.87	184.40	240.13	90.67	178.93	516.60	113.97	227.70
	Time(s)	0.056	0.121	0.218	0.131	0.116	0.889	0.085	0.546
	Ratio(%)	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

为了测试 ICPSO 与 IBPSO 在动态问题上的优势,在这里选用表 4.7 中四个典型动态优化函数进行实验,实验结果如表 4.10 所示。从表 4.10 中可以得出,4 个 Shift 动态问题测试证明 ICPSO 算法的均值、均方差都优于 IBPSO 算法,表明 ICPSO 算法在动态优化问题上其寻优精度与算法稳健性优于 IBPSO 算法。

表 4.10 ICPSO 与 IBPSO 算法的动态性能比较 (Shift 测试函数)

算法	指标	f_1	f_2	f_3	f_4
IBPSO	Mean	-449.84	-446.39	-179.91	-134.35
	Std.dev	0.336	1.864	0.061	2.318
ICPSO	Mean	-450	-450	-179.993	-140
	Std.dev	0	2.02e-05	9.89e-03	0

4.6 本章小结

本章将协同进化理论及免疫优化理论引入免疫粒子群进化算法中,阐述了免疫协同粒子群进化算法基本原理及其算法步骤。借鉴协同进化思想扩大了 PSO 算法搜索空间,促使粒子多方向性广泛搜索。人工免疫系统具有全局优化能力,能使粒子群逃离局部极值点。基准函数测试实验结果表明在收敛精度与全局搜索能力方面 ICPSO 算法优于最近几种变种粒子群算法。Shift 动态函数测试表明 ICPSO 算法具有较好的动态优化能力,证明该算法具有处理动态优化问题的优越性,适合于复杂非线性动态系统建模与辨识。

第5章

免疫蚁群自适应融合算法

前面几章研究了协同免疫进化算法、免疫双态粒子群算法以及免疫协同粒子群算法。蚁群算法也是仿生智能算法领域中的一个重要优化方法，本章将人工免疫系统与蚁群系统结合研究，融合两种算法的优势，研究了两类免疫蚁群自适应融合算法。第一类，针对免疫克隆选择设计相关搜索算子，构造局部最优搜索算子，利用自适应统计学习控制参数实现克隆选择与蚁群优化的有机结合。第二类，模拟自然界群体智能分工协作职能并结合免疫系统，研究了一种双态免疫优势蚁群算法，该算法将蚂蚁分成两种状态，扩大了解的搜索空间，将禁忌表中的抗体通过克隆扩增、高频变异等免疫算子操作得到精英蚂蚁，再对抗体记忆库引入局部最优免疫策略，有效抑制了收敛过程中的早熟停滞现象，并对这两类免疫蚁群算法进行理论分析与实验验证。同时免疫蚁群自适应融合算法的成功实现也为免疫系统与其他智能系统的自适应融合提供了指导思路。

5.1 引言

蚁群算法是 Dorigo 等人提出的一种仿生进化算法^[167, 168]，是模拟蚁群在觅食过程中依赖信息素通信行为而演变出的社会群体智能，目前蚁群优化已经成功应用于多个 NP 难的组合优化问题求解。蚁群算法的正反馈机制确保了最优化过程的快速性，具有贪婪启发式的搜索特征，易在算法搜索早期获得问题可接受解，然而在算法的执行后期，易出现早熟、停滞现象。

人工免疫算法具有快速随机的全局搜索能力,但对于系统中的反馈信息利用不足,往往做大量无为的冗余迭代,求解效率低。蚁群算法具有分布式并行全局搜索能力,通过信息素的积累和更新收敛于最优路径上,算法运行前期收敛较快,但容易陷入局部极值。人们利用蚁群算法信息素机制对算法寻优具有导向性,可以加强其他仿生智能算法的收敛性。Ahuja 在文献[5]提出了基于信息素的人工免疫变异方法,通过蚁群信息素引导免疫变异提高了人工免疫算法的寻优效率,进一步增强了人工免疫算法的收敛速度,避免了算法做无谓的冗余迭代。焦李成等人在文献[14]中利用蚁群信息素的无导师学习机制并结合免疫克隆选择机制,构造出一种基于信息素模因的免疫克隆选择算法,该算法无须提供确定候选的局部搜索策略,将免疫克隆选择算法与信息素机理结合研究,可以借鉴抗体种群的进化经验来提升算法的寻优效率,降低算法搜索代价。Ashish 等人提出了一种混合免疫蚁群算法^[5],运用免疫算法多样性及超变异能力来指导蚁群信息素生成与更新,经过免疫算法操作后的信息素来指导蚁群算法寻优,提高了算法搜索能力。Ling Q 等人在文献[70]中通过模拟生物免疫系统来保证蚁群算法的多样性,该算法引入免疫选择、免疫记忆、免疫代谢、密度控制等机制到蚁群系统,实验表明该算法收敛速度快,具有较强的寻优速度,能有效避免算法早熟。文献[169]在蚁群算法中引入了免疫疫苗思想,有效地克服了基本蚁群算法容易出现停滞现象的缺陷。然而现有关免疫系统与蚁群系统的研究一方面还只是借用免疫系统中的一些概念和免疫细胞的某些活动规律来改造现行方法,基本上没有突破“局部改造现行方法”的设计思路^[170];另一方面没有充分考虑两种算法的内在联系与本质区别。

为了充分汲取这两种算法的优点,利用克隆选择算法(CSA)与蚁群算法(ACO)各自的优势,本章将研究一种新的优化方法——基于抗体片段局部最优搜索的克隆选择和蚁群自适应融合算法(ACALA)。引入混沌扰动增加抗体种群的多样性以提高蚁群算法的搜索能力,利用克隆扩增、免疫基因等相关算子的操作增强克隆选择算法搜索的效率,通过自适应控制参数实现克隆选择与蚁群优化的优势互补的有机结合及局部最优搜索策略的应用,达到优化性能与时间性能的双赢。实验结果表明:该算法在收敛速度与求解精度上均取得了较好的效果。

5.2 蚁群算法优化机理

蚁群算法^[167]受真实蚁群觅食行为策略的启发, 是对自然界蚂蚁的寻径方式进行模拟而演变出的一种仿生进化算法。蚂蚁在运动过程中, 能够在它所经过的路径上释放一种称为外激素 (Pheromone) 的物质进行信息传递, 而且蚂蚁在运动过程中能够感知这种信息素, 并以此指导自己的行径方向, 自然界大量蚂蚁构成的蚁群集体表现出一种信息正反馈智能行为现象: 当某一路径上走过的蚂蚁越多, 即释放信息素越多, 则后来者选择该路径的概率就越大, 从而找到一条巢穴到食物源之间的最短路径。

为模拟蚂蚁实际智能行为, 设: m' 是蚁群中蚂蚁的数量, d_{ij} 是城市 i 转移到城市 j 之间的距离, η_{ij} 是路径 (i,j) 的能见度, $\eta_{ij} = 1/d_{ij}$, 反应由城市 i 转移到城市 j 的启发程度, $\tau_{ij}(t)$ 是在 t 时刻在路径 (i,j) 上信息量, p_{ij}^k 是蚂蚁 k 从城市 i 转移到城市 j 的状态转移概率, j 是尚未访问的城市节点, 状态转移概率 p_{ij}^k 如下所示。

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \in \text{allowed}_k} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)}, & s \in \text{allowed}_k \\ 0, & \text{否则} \end{cases} \quad (5.1)$$

式 (5.1) 中, $\text{allowed}_k = \{0, 1, \dots, n-1\} - \text{tabu}_k$ 表示蚂蚁 k 下一步允许选择的的城市; α 为信息启发式因子, β 为期望启发式因子, α 和 β 分别反应了蚂蚁在寻径过程中所积累的信息量及启发信息在蚁群选择路径中的相对重要性^[168]。为每只蚂蚁设计一个禁忌表 $\text{tabu}_k (k=1, 2, \dots, m)$, 该禁忌表记录 t 时刻蚂蚁 k 已走过城市节点号, 且不允许该蚂蚁在本次循环中重复经过该节点。本次循环结束后禁忌表被清空。

蚂蚁执行一次循环, 对各路径上的信息素按如下方式进行更新:

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \Delta \tau_{ij} \quad (5.2)$$

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (5.3)$$

对于单位长度信息素的数量 $\Delta\tau_{ij}$ 可以用不同的形式描述。Dorig 提出了三种不同的蚁群系统模型^[167], 分别称为 ant-cycle system (蚁圈)、ant-quantity system (蚁量) 以及 ant-density system (蚁密)。

① ant-cycle system 模型为

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{如果蚂蚁 } k \text{ 在本次循环中经过路径 } (i, j) \\ 0, & \text{否则} \end{cases} \quad (5.4)$$

② ant-quantity system 模型为

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{d_{ij}}, & \text{蚂蚁 } k \text{ 在时刻 } t \text{ 和 } (t+1) \text{ 经过边 } ij \\ 0, & \text{否则} \end{cases} \quad (5.5)$$

③ ant-density system 模型为

$$\Delta\tau_{ij}^k = \begin{cases} Q, & \text{蚂蚁 } k \text{ 在时刻 } t \text{ 和 } (t+1) \text{ 经过边 } ij \\ 0, & \text{否则} \end{cases} \quad (5.6)$$

式 (5.2) ~ (5.6) 中 L_k 为当前全局最优解的路径长度, Q 为常量, ρ 为信息素挥发系数, $(1-\rho)$ 表示信息素残留因子。 $\Delta\tau_{ij}$ 为本次循环中路径 (i, j) 上的信息素增量。 $\Delta\tau_{ij}^k$ 是第 k 只蚂蚁在本次循环中路径 (i, j) 上信息素增量。三种模型的区别在于 ant-cycle system (蚁圈) 应用全局信息, ant-quantity system (蚁量) 和 ant-density system (蚁密) 应用局部信息。求解 TSP 等组合优化问题时, 一般取 ant-cycle system 效果较好, 研究人员一般将 ant-cycle system 作为蚁群算法的标准模型。

5.3 基于统计学习的免疫融合蚁群算法 (ACALA)

5.3.1 带混沌扰动算子启发式蚁群算法

由上述可知, 蚁群算法的运动规则主要依据信息正反馈原理和启发式相结合。该算法将蚂蚁的搜索行为集中到最优解的附近可以提高解的质量和收敛速度。但这种搜索方式会使早熟收敛行为更容易发生。如果随机产生的某一路径信息量变化过快, 很容易出现搜索停滞现象, 控制信息量变化速度的一种改进方法是利用混沌运动的遍历性, 选择下一个被访问的城市, 以避免

蚁群搜索的停滞。这里采用 Logistic 混沌序列^[153]。

$$U_{n+1} = \mu U_n (1 - U_n) \quad (5.7)$$

式中, U_{n+1} 为混沌变量, $n=0,1,2,\dots$, $0 < U_0 < 1$; μ 为混沌系统的状态控制参数, 已证明^[153], 当 $\mu=4$ 时, 上式表示的系统完全处于混沌状态, U_{n+1} 在 $(0,1)$ 范围内遍历。

在蚁群中以小概率 P'_m 选择出 m' 个蚂蚁进行混沌搜索, 其中蚂蚁从第 i 个节点转移到第 j 个节点时, j 的位置选取可按式确定:

$$j = \text{round}(1 + U_{r+1} M_{r+1}) \quad (5.8)$$

式中 M_{r+1} 为剩下城市的个数。这样适时引入混沌扰动算子, 增加解的多样性, 避免出现进化停滞现象。其他按式 (5.1) 计算。

5.3.2 MAX-MIN 机制

蚁群算法选择下一解元素的概率, 它直接依赖于信息素轨迹和启发信息。以往的信息素更新机制, 在每个选择点上一个选择的信息素轨迹量明显高于其他的选择, 这样有可能导致搜索的停滞。通过限制信息素的轨迹区间, 可以有效地避免各信息素轨迹之间的差异过大。为了避免早熟现象, 引入 MAX-MIN 机制^[70], 即一个循环中只有路径最短的蚂蚁才有权限修改本条路径的 τ_{ij} , 将之限制在 $[t_{\min}, t_{\max}]$ 。其他路径则不予修改相应的 τ_{ij} 。这样就可以避免抗体群被少数亲和度最高的抗体占满以增强较优秀抗体实现亲和度成熟的机会, 即一些个体有可能经过亲和度成熟成为全局最优解, 以免陷入局部极值。修改策略在信息素更新之后, 最短路径的 $\tau_{ij}(t)$ 按照下式进行阈值判断选择。

$$\tau_{ij}(t+n) = \begin{cases} t_{\min}, & \tau_{ij}(t) \leq t_{\min} \\ \tau_{ij}(t), & t_{\min} \leq \tau_{ij}(t) \leq t_{\max} \\ t_{\max}, & \tau_{ij}(t) > t_{\max} \end{cases} \quad (5.9)$$

$$\begin{aligned} t_{\max(t)} &= \frac{1}{1-\rho} \cdot \frac{1}{J^{\text{opt}(t-1)}} \cdot m' \\ t_{\min(t)} &= \frac{t_{\max(t)}(1 - \sqrt[n]{\text{Pbest}})}{(\text{avg} - 1)\sqrt[n]{\text{Pbest}}} \end{aligned} \quad (5.10)$$

式 (5.10) 中 $\text{avg}=n/2$, n 为城市规模。 J^{opt} 为全局最优解, $\text{Pbest} \in (0,1)$ 是一个动态调节信息素的控制参数, 本章算法中取 0.5, m' 为蚂蚁的个数。

5.3.3 免疫操作搜索算子

1. 抗体片段的局部最优搜索算子

通过限制搜索的抗体片段长度范围, 利用高效的搜索算子对抗体片段内的基因位进行优化, 这样极大地减少了计算时间, 提高了搜索效率。下面我们结合 TSP 具体问题, 讨论抗体片段的局部最优搜索算子。

设抗体片段为 $V=(v_1, v_2, \dots, v_p)$, e_1 初始集合为 (v_1, v_p) , e_2 初始集合为 $(v_2, v_3, \dots, v_{p-1})$, $v_i, v_{i+1} \in e_1$, v_j 是 e_2 中的第一点, $\text{LA}=\{(\Delta d_i, v_i, v_j, v_{i+1})\}$ 为抗体片段局部最优搜索算子。 $\Delta d_i, v_i, v_j, v_{i+1}$ 满足下式:

$$\Delta d_i = \min(d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1})) \quad (5.11)$$

抗体片段局部最优搜索算子

抗体片段局部最优搜索算子 (LS) 步骤描述:

```
begin:
{
while ( $|e_1| < |V|$ ) do{
    抗体片段局部最优搜索算子  $\text{LA}=\{(\Delta d_i, v_i, v_j, v_{i+1})\}$ ;
 $\Delta d_i = \min(d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1}))$ ;  $v_i, v_{i+1} \in e_1, v_j \in e_2$ ;
    当  $v_i$  取  $e_1$  中最后一个元素时,  $d(v_i, v_j) + d(v_j, v_{i+1}) - d(v_i, v_{i+1})$  就退化为  $d(v_i, v_j)$ 
    将  $v_j$  插入  $v_i$  之后;  $e_1 = e_1 \cup v_j$ 
    从  $e_2$  中移出  $v_j$ ;  $e_2 = e_2 - v_j$ 
}
}
end
```

2. 基于抗体片段的最小生成树搜索算子

最小生成树 (MST), 设 $N=(V, E, C)$ 是一个无回路且连接所有顶点的无向加权连通图, 其中 V 是顶点, E 是边数, C 表示该图的权图, U 是 V 的一个

非空子集,若 $u、v$ 满足 $\text{Weight}(u,v)=\min\{\text{weight}(u_0,v_0)|u_0\in U,v_0\in V-U\}$,则必存在 N 的一棵最小生成树 (MST),该生成树包括边 (u,v) 。

设抗体片段为 (u_1, u_2, \dots, u_p) , 其 u_1, u_2, \dots, u_p 是顶点, p 是边数, C 表示该图的权图,权值为两点间的欧几里得距离。采用 MST-Prim 算法构造最小生成树,再通过绕树两周算法,得到基于最小生成树的一条最优遍历的抗体片段。

由表 5.1 实验证明可得:基于抗体片段的局部最优搜索算法 (ACALA) 性能优于基于最小生成树的局部最优搜索算法 (ACAMST),故本章采用基于抗体片段的局部最优搜索算子。

表 5.1 ACAMST 与 ACALA 的比较 (测试 10 次)

问题 (S_0)	方法	最长	最短	平均	θ (%)
att48 (33524)	ACAMST	33607	33523.7085	33561.8276	0.11
	ACALA	33523.7085	33523.7085	33523.7085	0
Eil101 (642.3)	ACAMST	659.9689	645.1925	653.0580	1.67
	ACALA	652.7937	640.2116	644.9200	0.41

说明: 平均百分误差 $\theta = \frac{\sum_{i=1}^{10} (S_{T_i} - S_0)}{10S_0} \times 100\%$, 其中 S_{T_i} 是第 i 次的最短路径, S_0 是已知最短路径。

3. 免疫疫苗操作

免疫疫苗用局部优势抗体片段特征信息以一定的强度干预全局搜索行为,抑制或避免算法求解过程中的一些重复和无效的工作,以克服原进化算法中随机搜索算子操作的盲目性。在算法执行过程中,可以运用免疫疫苗有针对性地抑制抗体群进化中出现的一些退化现象,促使群体适应度相对稳定地提高,从而加快算法的收敛速度。

免疫疫苗的提取:基于抗体片段的局部最优搜索时,将比较优秀的抗体片段信息保留存入疫苗库,在进化的过程中,不断地更新疫苗库,免疫疫苗的提取不会有太大的时间开销。为免疫疫苗注射提供注射疫苗,而构造疫苗的时间复杂度是线性的。对接种的抗体随机地从疫苗库 $V(t)$ 中抽取一个疫苗 (疫苗组) V 。以一定的概率接种疫苗,没有将疫苗接种作为算法的一个独立

步骤。原因在于免疫疫苗比 Inver_over 算子启发性更强, 种群多样性不够时, 免疫疫苗操作会加速多样性损失, 导致算法早熟收敛, 为了折中抗体免疫疫苗以一定的概率进行操作。

```

Begin
for  $Y(K)$  中的每个抗体  $Y_i(K)$  do
{
    if (Rand()  $\leq P_v$ )// 其中  $P_v$  为疫苗接种概率;
        Vaccination ( $Y_i(K)$ );
    else
        Inver_over ( $Y_i(K), P$ ); //  $P$  为变异概率;
}
end

```

设对抗体 A 的接种疫苗操作 Vaccination (A), 具体如下。

```

Vaccination ( $A$ )
{
     $A' = A$ ;
    从疫苗库  $V(t)$  中随机选择一个疫苗 (疫苗组)  $V$ ;
     $C$  为  $V$  中的第一个城市,  $C'$  为  $V$  中  $C$  的最后一个城市;
     $i=1, j=i+|V|-1$ ; // ( $|V|$  表示疫苗基因  $V$  的城市个数)
    do {if( $A(i) == C \&\& A(j) == C' \&\& |A(i:j)| == |V|$ )
        then 将  $V$  插入  $A(i)$  之前, 得到中间抗体  $A^{\text{mid}}$  在不包含  $V$  的  $A^{\text{mid}}$  中, 删除与  $V$  相同的城市号得到新的抗体  $A^{\text{new}}$ 
         $i=i+1; j=j+|V|-1$ ;
    } while( $i < |A| - |V| + 1$ )
    if fitness( $A^{\text{new}}$ ) > fitness( $A'$ ) then  $A = A'$ ;
}

```

定义 5.1: 免疫基因操作 T_g^C 主要包括变异及疫苗注射, 即 $Z(K) = T_g^C(Y(k))$ 。其中变异算子基于 Inver_over 操作算子^[53], 变异概率如下:

$$p = p \times (1 - t \times 0.01 / T) \quad (5.12)$$

式中, P 为变异概率, t 为当前进化代数, T 为总的进化代数。

定义 5.2: 免疫选择 T_s^C , $S_i(k)=T_s^C(Z(K))=\max\{\text{affinity}(Z_i(k))\}$, 然后, $S_i(k) \cup A_i(k) \rightarrow A_i(k+1)$, 这样通过局部择优实现了种群的压缩, 同时保证了抗体群中的最优解不会变差。

5.3.4 基于统计学习的自适应免疫蚁群融合算法

1. 控制参数 P_g 自适应调节策略

为了充分汲取 CSA 与 ACO 两种算法的优势, 合理地分配两者在算法执行过程中的权重, 本章算法采用了控制参数 P_g 自适应调节策略。

$$P_g = F(T) = \int_{-\infty}^T f(x) dx \quad (5.13)$$

其中, $f(x)$ 是概率密度函数, 公式如下:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \quad (5.14)$$

式 (5.14) 中 σ 是标准方差, μ 是数学期望值。当变量 T 等于 μ 时, $P_g=0.5$, 此时意味着 ICSA 与 ACO 有均等概率执行。式 (5.13) 中 T 更新规则如下:

① 当 CSA 算法操作本代产生的最优解优于上代时, 则说明 CSA 算法占有优势, $T=T+\sigma/C$; 其中 C 为一个常量, 在本书算法中 C 取 500。

② 当 ACO 算法操作本代产生的最优解优于上代时, 则说明 ACO 算法占有优势, $T=T-\sigma/C$ 。

这种自适应的参数调节通过竞争机制的引入, 更好地促进了算法的融合。 σ 、 μ 的取值不会影响算法的效果, 为了简单, 我们就取标准正态分布。

2. ACALA 算法流程

在自适应融合算法 (ACALA) 算法中免疫克隆选择算法与蚁群优化算法都对同一个种群进行操作, CSA 产生新的解更新种群, 通过抗体片段局部最优搜索更好地促使种群亲和度提高, 而种群的亲和度提高反过来改变信息素, 从而提高 ACO 解的质量。另外, ACO 产生优质的解更新种群, 通过抗体片段局部最优搜索进一步提高种群的质量, 更有利于 CSA 产生更好的解, 从而快速地找到最优解, 汲取三种算法的精华, 加快算法的收敛速度, 提高

了求解精度。

ACALA 算法流程执行步骤如下。

ACALA 算法流程执行步骤描述：

begin

步骤 1: 随机产生初始抗体种群 P , 记忆库 M , 信息素 P_h , P_g , α , β 等参数初始化。

步骤 2: CSA 与 ACO 竞争, 获胜者拥有对种群的操作权。

步骤 3: while(终止条件不满足)do{

步骤 4: if(bApplyCSA(P_g)==true)

步骤 5: {计算 P 中所有抗体亲和度, 选择 K 个高亲和度的抗体克隆得到 $Y(k)$ 。

步骤 6: 对克隆后的抗体群 $Y(k)$ 进行免疫基因操作。

步骤 7: 计算每个抗体的亲和度, 得到一个当代最优解 S 。}

步骤 8: else{

步骤 9: 将 m' 只蚂蚁置于城市规模为 n 的各起始节点。

步骤 10: $m' P_m$ 按照式 (5.8) 进行混沌遍历, $m'(1 - P_m)$ 按照式 (5.1) 计算转移概率, 根据概率移动每只蚂蚁到下一节点。

步骤 11: m' 只蚂蚁遍历 n 个节点后, 路径最短的蚂蚁才有权修改 $\tau_{ij}(t)$, 按照式 (5.9)。

步骤 12: 所有路径信息素按照式 (5.2) 更新。

步骤 13: 计算每个抗体的亲和度, 得到一个当代最优解 S 。}

步骤 14: 对 S 进行基于抗体片段的局部最优搜索。

步骤 15: P_g 自适应调节, 按照式 (5.13)。

步骤 16: 更新记忆库 M 。}

步骤 17: 输出最优解 S , 结束。

end

5.3.5 ACALA 算法收敛性分析及实验仿真

1. ACALA 算法收敛性分析

引理 5.1: 文献[171]研究表明蚁群优化算法 $\{\tau(m+1), x(m+1), f^*(m+1)\}$ 仅与 $\{\tau(m), x(m), f^*(m)\}$ 有关, 而与搜索周期 m 无关, 并且是收敛的。

引理 5.2: 文献[8]研究表明基于免疫克隆选择策略算法的种群序列

$\{A(n), n \geq 0\}$ 是有限齐次马尔可夫链，并且是以概率 1 收敛的。

定理 5.1: 基于抗体片段局部最优搜索的克隆选择与蚁群自适应融合算法的种群序列 $\{A(n), n \geq 0\}$ 是有限齐次马尔可夫链。

证明：由引理 5.1、引理 5.2 可知，基于抗体片段局部最优搜索的克隆选择与蚁群自适应融合算法的种群序列 $\{A(n), n \geq 0\}$ 操作过程状态变化均在有限空间中进行。即有 $A(k+1) = T(A(k)) = T_s \circ T_m \circ T_c \circ T_d \Theta(A(k))$ ，其中 $\Theta, T_s, T_m, T_c, T_d$ 分别代表蚁群优化、免疫选择、克隆、免疫基因操作、局部搜索， \circ 表示运算合成符号 $T_s, T_m, T_c, T_d, \Theta$ 均与 n 无关，因此 $A(k+1)$ 仅与 $A(k)$ 有关，即 $\{A(n), n \geq 1\}$ 是有限齐次马尔可夫链。证毕。

定理 5.2: 基于抗体片段局部最优搜索的克隆选择与蚁群的自适应融合算法是以概率 1 收敛的。

上述算法种群规模为 m ，通过自适应融合算法操作以后，得到优秀抗体群 m_0 。可将种群中的全部近似解看成是状态空间 S 个体， $S_i \in S$ 表示 S 中的第 i 个状态。用 M^* 表示由 $f(x)$ ($f(x)$ 是 X 上的适应度函数) 在 S 上取最优值的所有抗体组成， $M^* = \{x \in X \mid f(x) = \max f(x)\}$ 则可定义算法的收敛性如下：

$$\lim_{k \rightarrow \infty} \sum_{S_i \cap M^* \neq \Phi} P\{A_k^i\} = 1 \quad (5.15)$$

式 (5.15) 表明，当算法迭代到足够多的次数后，最优解集中包含全局最佳个体的概率接近于 1，称为算法收敛。

证明：设随机过程 $\{A(k)\}$ 的转移概率为 $p_{ij}(k)$ ， $I = \{i \mid S_i \cap M^* \neq \Phi\}$ ，

且 $p_{ij}(k) = p\{\frac{A_k^j}{A_k^i} \geq 0\} \geq 0$ ，记 $p\{A_k^i\}$ 为 p_i^k ， $p_k = \sum_{i \in I} p_i(k)$ ，则由马尔可夫

链的性质可知：

$$p_{k+1} = \sum_{S_i \in S} \sum_{j \in I} p_i(k) p_{ij}(k) = \sum_{i \in I} \sum_{j \notin I} p_i(k) p_{ij}(k) + \sum_{i \in I} \sum_{j \in I} p_i(k) p_{ij}(k)$$

因为

$$\sum_{i \in I} \sum_{j \in I} p_i(k) p_{ij}(k) + \sum_{i \notin I} \sum_{j \in I} p_i(k) p_{ij}(k) = \sum_{i \in I} p_{i(k)} = p_k$$

所以

$$\sum_{i \notin I} \sum_{j \notin I} p_i(k) p_{ij}(k) = p_k - \sum_{i \notin I} \sum_{j \in I} p_i(k) p_{ij}(k)$$

则

$$0 \leq p_{k+1} \leq p_k - \sum_{i \notin I} \sum_{j \in I} p_i(k) p_{ij}(k) \leq p_k \leq 1$$

于是有

$$\lim_{k \rightarrow \infty} p_k = 0$$

则

$$1 \geq \lim_{k \rightarrow \infty} \sum_{S_i \cap M^* \neq \emptyset} p_i(k) \geq \lim_{k \rightarrow \infty} \sum_{i \in I} p_i(k) = 1 - \lim_{k \rightarrow \infty} p_k = 1$$

可证明式 (5.15) 以概率 1 收敛。

2. ACALA 算法测试实验及分析

TSP 问题是著名的 NP 难问题, 有着广泛的应用^[133], 难于用常规算法求解。它经常也被用来作为各种非线性优化方法的评价标准。TSP 问题可以简单描述为: 已知 N 个城市 $v = \{v_1, v_2, v_3, \dots, v_N\}$, 以及任意两城市之间的距离 $d = (v_i, v_j)$, 求一条经过 v 中所有城市一次且仅一次的闭合 $v_\pi = \{v_{\pi(1)}, v_{\pi(2)}, \dots, v_{\pi(N)}\}$, 使得总行程 $d = \sum_{i=1}^{N-1} d(v_{\pi(i)}, v_{\pi(i+1)}) + d(v_{\pi(N)}, v_{\pi(1)})$ 最小。目前有关蚁群算法与人工免疫算法的融合用于求解 TSP 问题的还不多^[14]。

定义 5.3: 抗体编码, 抗体在人工免疫系统中一般指问题的候选解。在免疫算法求解 TSP 问题时, 抗体 A 对应候选路径, 通常采用实数编码 (用城市的序号表示)。

定义 5.4: 抗原定义, 抗原抽象为 TSP 问题的求解, 在所有可能的遍历 N 个城市的路径中选择一条最短的路径, 通过算法 ACALA 优化, 最终得到的最短路径就是抗原, 如下式所示:

$$\min f = \sum_{i=1}^{N-1} d(v_i, v_{i+1}) + d(v_N, v_1) \quad (5.16)$$

其中 f 是遍历所有城市的总距离。

定义 5.5: 亲和度定义, 抗体与抗原的亲和度反应抗体与抗原的匹配程度, 一般取路径距离的倒数, 为了更好区分抗体间的优劣, 亲和度函数定义如下式所示:

$$\text{affinity}(A_i) = \frac{F - \text{dist}(A_i)}{\sum_{j=1}^m (F - \text{dist}(A_j)) + \partial} \quad (5.17)$$

式中 $\text{dist}(A_i)$ 为第 i 个抗体遍历整个城市的距离, ∂ 为平滑系数, 以保证分母不为 0, F 为当代路径长度的最大值, 即 $F=\max(\text{dist}(A_1), \text{dist}(A_2), \dots, \text{dist}(A_n))$ 。

实验仿真环境: Windows XP 系统, AMD 处理器, 1.90GHz, 448MB 内存。仿真软件为 MATLAB 7.0。为了更好地说明该算法的有效性, 我们选用国际上通用的 TSPLIB 测试库中的多个实例进行测试, 实例 1 对 Rd100 (城市数为 100) 进行 10 次测试, 变异初始概率 $P_m=0.75$, 疫苗接种概率 $P_v=0.4$, 抗体种群规模 $m=100$, 蚁群数量为 50, $\alpha=1$, $\beta=5$, $Q=10$ 。蚁群混沌小概率 $p_l=0.1$, 克隆算子比例系数 $\alpha=300$, 克隆算子常数 $b=20$, 图 5.1 为针对 Rd100 实例的基于不同 P_g 值收敛过程的对比图。 $P_g=0$, 采用 ACO-LS 算法, 求得的路径距离最大为 8695.3094, 最小为 8132.3276, 均值为 8386.6809, 平均误差为 6.02%; $P_g=1$, 采用 CSA-LS 算法, 求得的路径距离最大为 8211.1782, 最小为 7958.8714, 均值为 8089.7097, 平均误差为 2.27%。自适应 P_g 采用自适应融合的 ACALA 算法, P_g 初始值设为 0.5, 求得的路径距离最大为 7944.3219, 最小为 7910.3962, 其中 8 次找到最优值, 均值为 7917.1814, 平均误差为 0.086%, 优于 TSPLIB 中提供的最优值 7910.4。从图 5.1 可以看出, ACO-LS 在初期收敛速度较快, 但后期收敛精度较低; CSA-LS 在初期得到的优化值较大, 但后期收敛精度较高; 而 ACALA 融合了 ACO-LS 和 CSA-LS 两者的优点, 收敛速度更快, 收敛精度更高。

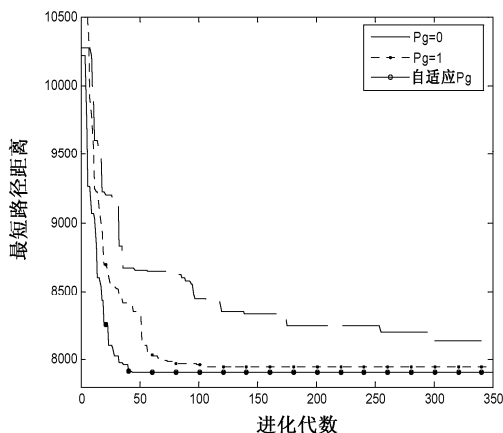


图 5.1 基于不同 P_g 值的收敛过程对比 (Rd100)

图 5.2 为自适应控制参数 P_g 变化曲线图, 该图表明, 在 $P_g=0.5$ 时, CSA 与 ACO 享有均等的机会对抗体群进行操作, 竞争获胜者将继续执行算法。在算法执行的初期 P_g 逐渐降低小于 0.5, 算法采用 ACO-LS, 这表明在算法执行的初期 ACO 的搜索能力强于 CSA; 但随着进化代数的进行, P_g 的值逐渐增加, 最终保持着较高的平稳值。当大于 0.5 时, 算法采用 CSA-LS, 这表明 CSA 搜索能力强于 ACO。Rd100 测试实例的最优路径如图 5.3 所示。

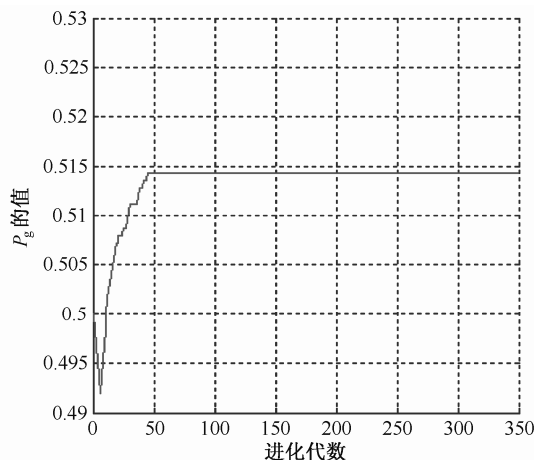


图 5.2 Adaptive P_g 变化曲线图

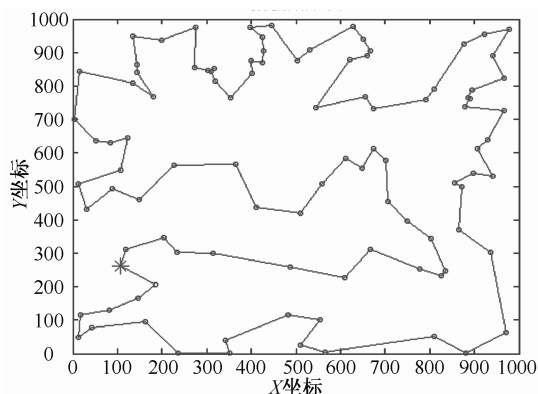


图 5.3 Rd100 的最优路径图

从表 5.2 可以看出 ACALA 融合了 ACO-LS 和 CSA-LS 两者的优点, 收

敛速度更快，收敛精度更高。

表 5.2 基于不同 P_g 值的其他测试实例（测试 10 次）

问题 (S_0)	方法	最长	最短	平均	$\theta(\%)$
St70 (678.598)	$P_g=0$	750.4558	700.7784	720.5562	6.18
	$P_g=1$	703.9066	677.1096	686.2392	1.13
	Adaptive P_g	691.0787	677.1096	679.4500	0.13
KroD100 (21294.3)	$P_g=0$	23000.6022	21908.7909	22443.8965	5.4
	$P_g=1$	22051.7421	21627.3227	21776.7934	2.27
	Adaptive P_g	21501.2133	21294.2908	21333.0259	0.18
Pr120 (1666.5)	$P_g=0$	1794.6597	1701.6968	1748.4324	4.92
	$P_g=1$	1716.9317	1641.9683	1663.2472	0.2
	Adaptive P_g	1630.2301	1615.8109	1622.0114	0
Gr202 (549.99)	$P_g=0$	540.6836	509.4948	520.8149	0
	$P_g=1$	528.17768	504.3370	512.1675	0
	Adaptive P_g	491.1264	487.5290	488.7385	0

表 5.3 中的 ICA 为本书第 2 章中提出的一种局部最优免疫优势克隆选择算法，IDIA 为文献[8]中基于优秀抗体的免疫算法，从表 5.3 中可以看出本章算法 ACALA 性能在 TSP 测试实例 Gr96、KroD100、Pr120、Gr202、Tsp225、Pcb442 等问题上均优于 IDIA 基于优秀抗体的免疫算法，同时在 TSP 测试实例 Gr96、KroD100、Pr120、Gr202、Tsp225 等问题上本章 ACALA 算法优化性能均优于局部最优免疫优势克隆选择算法 ICA，而在 Pcb442 问题上，ACALA 算法优化性能与 ICA 优化性能相当。

表 5.3 ACALA 与免疫算法比较（测试 10 次）

问题 (S_0)	方法	最长	最短	平均	$\theta(\%)$
Gr96 (512.31)	IDIA	527.26	511.50	516.98	0.91
	ICA	517.5182	510.8863	514.3886	0.41
	ACALA	514.2561	510.8863	512.3902	0.016
KroD100 (21294.3)	IDIA	22292.2	21294.3	21449.1	0.73
	ICA	21420.4324	21294.2908	21359.4767	0.31
	ACALA	21501.2133	21294.2908	21333.0259	0.18

续表

问题 (S_0)	方法	最长	最短	平均	$\theta(\%)$
Pr120 (1666.5)	IDIA	1699.4	1657.6	1675.5	0.54
	ICA	1654.9390	1614.8899	1635.3732	0
	ACALA	1630.2301	1615.8109	1622.0114	0
Gr202 (549.99)	IDIA	501.3	490.67	495.67	0
	ICA	496.21	487.6392	492.0620	0
	ACALA	491.1264	487.5290	488.7385	0
Tsp225 (3859)	IDIA	4144.2	3898.4	4020.5	4.18
	ICA	3997.42	3925.69	3954.37	2.47
	ACALA	3981.2547	3895.5250	3925.0823	1.71
Pcb442 (5078.35)	IDIA	5468.83	5184.49	5278.40	3.94
	ICA	5276.6	5118.0	5193.0	2.26
	ACALA	5276.6	5118.0	5193.0	2.26

表 5.4 中 ACS 为文献[172]中的改进蚁群算法,从表 5.4 可以得出 ACALA 的各项优化指标性能明显优于文献[172]中的改进蚁群算法。

实验分析:从图 5.1、表 5.3 和表 5.4 可以得出,ACALA 融合了 ACO-LS 和 CSA-LS 两者的优点,形成优势互补。其求解精度、优化性能优于改进的免疫克隆选择算法(如 ICA、IDIA),同时也优于改进的蚁群算法(ACS),其优化性能汲取了免疫克隆选择算法与蚁群优化算法的优势,明显强于原有算法优化性能。

表 5.4 ACALA 与 ACS 的比较(测试 10 次)

问题 (S_0)	方法	最长	最短	平均	N^{best}
Eil51 (428.87)	ACS	437.80	428.87	429.48	4
	ACALA	435.66	428.87	430.15	5
Eil76 (544.37)	ACS	565.16	545.97	553.77	0
	ACALA	556.33	544.37	550.04	2
KroA100 (21285.44)	ACS	22499.74	21285.44	21456.98	7
	ACALA	21307.42	21285.44	21289.84	8
D198 (15808.65)	ACS	16936.53	15919.55	16143.66	0
	ACALA	15892.78	15825.53	15847.12	0

从上述实验结果中得到启示, ACALA 算法的成功自适应融合同时也作为免疫克隆选择算法与其他仿生智能优化算法(如遗传算法、粒子群算法、人工鱼群算法等)的自适应融合提供了指导性思路, 将加快推进人工免疫系统与其他智能算法的有机结合研究。

5.4 免疫双态蚁群融合算法(BAIA)

5.4.1 双态蚁群机理

“双态性”指蚁群社会所具有两种不同类型信息素调节机制的蚁群。基本的蚁群算法在搜索过程中主要是利用了正反馈原理。然而该过程在强化性能较好解的同时, 却容易导致停滞现象。算法在寻优过程中, 很大程度上受早期发现的较好解的影响, 这些较好解以极大的概率引导蚁群走向局部最优点, 这正是蚁群算法信息素容易出现停滞现象的主要原因。解决蚁群算法迭代过程中停滞现象和保证其具有较强的探索能力的方法就是应在“探索”和“利用”之间建立一个平衡点。对蚁群社会行为的研究表明, 真实蚁群社会中的所有蚂蚁各司其职, 又相互依赖协作, 自然形成一个有机的整体。在执行某项任务时, 个体之间通过各自分泌的信息素相互间通信。不同种类的蚁群将有不同的信息素调控机制, 这样就更能够模拟蚁群社会中的蚂蚁有组织, 有分工的特点, 这种分工组织方式对蚁群完成复杂任务具有十分重要的作用。因此可以将具有 m 只蚂蚁的蚁群分成两种不同行为特征子蚁群, 其比例设为: $r_1:r_2$ 。

蚂蚁状态 1: 蚂蚁以确定性和状态转移概率性相结合选择下一个要到达的城市。

$$j = \begin{cases} \arg \max_{j \in \text{allowed}_k} \{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)\}, & q < q_0 \\ p_{ij}^k, & \text{否则} \end{cases} \quad (5.18)$$

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{s \in \text{allowed}_k} \tau_{is}^\alpha(t) \eta_{is}^\beta(t)}, & s \in \text{allowed}_k \\ 0, & \text{否则} \end{cases} \quad (5.19)$$

m 是蚁群中蚂蚁的数量, d_{ij} 是城市 i 到城市 j 之间的距离, η_{ij} 是边 (i, j) 的能见度, $\eta_{ij}=1/d_{ij}$, 反映由城市 i 转移到城市 j 的启发程度, $\tau_{ij}(t)$ 是在 t 时刻边 (i, j) 上的信息量, p_{ij}^k 是蚂蚁 k 从城市 i 转移到城市 j 的状态转移概率, j 是尚未访问的城市, 则状态转移概率 p_{ij}^k 式中, $\text{allowed}_k=\{0,1,\cdots,n-1\}-\text{tabu}_k$ 表示蚂蚁 k 下一步允许选择的城市。 α 为信息启发式因子, β 为期望启发式因子, 分别反映了蚂蚁在运动过程中积累的信息和启发信息在蚂蚁选择路径中的相对重要性。为每只蚂蚁设计一个禁忌表 $\text{tabu}_k(k=1,2,\cdots,m)$, 记录在 t 时刻蚂蚁 k 已走过的城市, 不允许该蚂蚁在本次循环中重复经过。本次循环结束后禁忌表被清空。

蚂蚁状态 2: 蚂蚁确定性和混沌遍历性相结合选择下一个要到达的城市。

$$j = \begin{cases} \arg \max_{j \in \text{allowed}_k} \{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)\}, & q < q_0 \\ \text{round}(1 + U_{r+1} M_{r+1}), & \text{否则} \end{cases} \quad (5.20)$$

M_{r+1} 为剩下城市的个数, U_{r+1} 为混沌序列 Logistic 映射。这样适时引入混沌扰动算子, 增加解的多样性, 避免出现进化停滞现象。

自适应参数 q_0 的选取如下:

$$q_0(t+1) = \begin{cases} 0.95q_0(t), & 0.95q_0(t) \geq q_{0\min} \\ q_{0\min}, & \text{否则} \end{cases} \quad (5.21)$$

根据式 (5.21) 可看出, 当 $q < q_0$ 时, 算法是采用确定性搜索, 当 $q \geq q_0$ 时, 算法等同于随机搜索, 在算法迭代的初期 q_0 选取较大的初始值, 以较大的概率进行确定性搜索, 这样可以充分利用问题本身的特征 (两点间距离), 加快寻找局部较优路径的速度; 在算法的中后期 q_0 选取较小的值, 增大随机搜索的概率, 从而扩大搜索空间寻找全局最优解。

5.4.2 信息素的更新规则

蚂蚁完成一次循环，对各路径上的信息素进行更新：

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta \tau_{ij} \quad (5.22)$$

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k \quad (5.23)$$

$$\Delta \tau_{ij}^k = \begin{cases} Q/L_k + H^* \text{affinity}(k), & \text{第 } k \text{ 只蚂蚁在本次循环中经过}(i, j) \\ 0, & \text{否则} \end{cases} \quad (5.24)$$

式中 L_k ——当前第 k 只蚂蚁走过的路径长度， Q ——常量， ρ ——信息素挥发系数， $1-\rho$ 表示信息素残留因子。 $\Delta \tau_{ij}$ 为本次循环中边 (i, j) 上信息素增量。 $\Delta \tau_{ij}^k$ 是第 k 只蚂蚁在本次循环中边 (i, j) 上信息素增量， m 为蚂蚁的数量。 $\text{affinity}(k)$ 为第 k 只蚂蚁的亲和度。

$$H = \frac{Q}{L_k \cdot \text{affinity}_{\text{avg}}} \quad (5.25)$$

信息素初始化 $\tau_{ij}(0) = C/(n^* S_0)$ ， n 为城市的节点数， C 为常量。 S_0 为已知最优值。信息素的规范化如下：

$$\tau_{ij} = \tau_0 (1 + \log(\tau_{ij} / \tau_0)) \quad (5.26)$$

式 (5.26) 能够将 τ_{ij} 相应予以压缩。 τ_{ij} 值越小，压缩越少； τ_{ij} 值越大，压缩越大。经过式 (5.26) 压缩以后，各边间的信息素差距大为减小，为下一代各只蚂蚁提供较为均等的机会。这样就可以增强较优秀抗体实现亲和度成熟的机会，同时避免了抗体群被少数亲和度最高的抗体占满，导致那些有可能经过亲和度成熟过程成为全局最优解的抗体受到遏制，从而使算法陷入局部极值。

5.4.3 BAIA 算法流程

将禁忌表中每只蚂蚁走过的路径视为一个抗体，将当代抗体群中的路径长度最短的抗体看作抗原。BAIA 算法流程执行步骤如下。

BAIA 算法 ():

Step1: 蚁群参数初始化, 信息素矩阵初始化。

Step2: 生成 m 只蚂蚁, 放到 n 个城市结点上, 其中 $m * r_1$ 只蚂蚁按状态 1 行动, $m * r_2$ 只蚂蚁按状态 2 行动。

Step3: for 每只蚂蚁 do

Repeat

按自己的行为规则选择下一城市;

移动到下一城市;

将该城市的索引加入自己的禁忌表、

Until 不能再向前移动

End for

Step4: 视禁忌表为一个抗体群, 选择 k 个高亲和力抗体, 按照亲和力及对应的位置克隆扩增: $A'(K) = T_c^C(A(k))$ 。

Step5: 采用 Inver_over 算子, 进行高频变异 $Z(K) = \text{Inver_over}(A'(K))$ 。

Step6: 计算亲和力进行免疫选择得到新的抗体群。 $S_i(k) = T_s^C(Z(K)) = \max \{\text{affinity}(Z_i(k))\}$ 实现信息交换, 更新信息素。

Step7: 蚂蚁群体的进化。保留精英蚂蚁进入抗体记忆库 M 。对抗体记忆库进行局部最优免疫优势操作: $M'(k) = \text{LOIO}(M(k))$ 。

Step8: 结束条件。判断结束条件 (结束条件为最大迭代次数), 如果满足结束条件, 输出最优个体, 否则清空所有蚂蚁的禁忌表, 转第 (3) 步。

5.4.4 BAIA 算法性能分析

1. BAIA 收敛性分析

引理 5.3: 文献[171]研究表明蚁群优化算法 $\{\tau(n+1), x(n+1), f^*(n+1)\}$ 仅与 $\{\tau(n), x(n), f^*(n)\}$ 有关, 而与搜索周期 n 无关, 并且是收敛的。

引理 5.4: 文献[8]研究表明基于免疫克隆选择策略算法的种群序列 $\{A(n), n \geq 0\}$ 是有限齐次马尔可夫链, 并且是以概率 1 收敛的。

定理 5.3: 免疫—蚁群算法的优化解序列 $\{x(n), n \geq 0\}$ 是有限齐次马尔可

夫链。

证明：由引理 5.4，基于免疫克隆选择策略算法的种群序列 $\{A(n), n \geq 0\}$ 是有限齐次马尔可夫链， $A(n+1)$ 仅与 $A(n)$ 有关，与迭代次数 n 无关；由引理 5.1， $\{\tau(m+1), x(m+1), f^*(m+1)\}$ 仅与 $\{\tau(m), x(m), f^*(m)\}$ 有关，而与搜索周期 m 无关。同时，由免疫蚁群模型可知，每次迭代仅更新 Ant-Cycle 模型中的最优信息素。

$x(n+1) = (x_i(n+1) = T_g^i \circ T_m^i \circ T_c^i \circ T_s^i(x(n)), (i \leq N-1); x_N(n+1) = x_{i0}(n)) (T_g^i, T_m^i, T_c^i, T_s^i$ 分别表示免疫优势、克隆扩增、克隆变异、信息素更新算子)

式中， $i_0 = \operatorname{argmin} \{f(x_j(n))\}$ 表示取亲和度最优的个体为 $x_{j_0}(n)$ ，且转移概率矩阵为：

$$p\{x, y\} = p\left\{\begin{array}{l} x(n+1) = y \\ x(n) = x \end{array}\right\} = \begin{cases} \prod_{k=1}^N P\{T(x(n))_k = y_k\}, \exists i_0 M(x), \text{使 } y_N = x_{i_0} \\ 0, \text{ 否则} \end{cases}$$

上式表明， $x(n+1)$ 仅与 $x(n)$ 有关，而与 n 无关。免疫—蚁群算法的优化解序列 $\{x(n); n \geq 0\}$ 是有限齐次马尔可夫链。

定理 5.4：免疫—蚁群算法的优化解马尔可夫序列以数值为 1 的概率收敛到满意解集 B 的子集 B_0^* ，这里定义 $B_0^* = \{y = (y_1, \dots, y_N); y_N \in M\}$ ，即有

$$\lim_{n \rightarrow \infty} p\left\{\frac{x(n) \in B_0^*}{x(0) = x_0}\right\} = 1 \quad (5.27)$$

证明：设 x' 是 $f(x)$ 的唯一最小解，由引理 5.4 和定理 5.3 知 $P\{x, y\}$ 有如下性质：

- 1) 当 $x, y \in B_0^*$ 时，有 $P\{x, y\} > 0$ ， $P\{y, x\} > 0$ ，即 $x \leftrightarrow y$ ；
- 2) 当 $x \in B_0^*$ ， $y \notin B_0^*$ 时，有 $P\{y, x\} = 0$ ，即 $x \rightarrow y$ 。

因此， B_0^* 为正常返的非周期不可约闭集，

$S^N \setminus B_0^*$ 为非常返的状态集： $\lim_{n \rightarrow \infty} p\left\{\frac{x(n) = y}{x(0) = x_0}\right\} = \begin{cases} \Pi(y), y \in B_0^* \\ 0, y \notin B_0^* \end{cases}$

故有 $\lim_{n \rightarrow \infty} p\left\{\frac{x(n) \in B_0^*}{x(0) = x_0}\right\} = 1$

2. BAIA 时间复杂度分析

蚁群的时间复杂度 $T(n) = O(N_C \cdot m \cdot n^2)$ (m 为蚂蚁的个数, n 为城市个数, N_C 为循环次数), 免疫优势的时间复杂度 $T(n) = O(N_C \cdot m \cdot n^2 \log(n))$ (m 为抗体个数)。免疫选择、变异等其他操作的最坏时间复杂度为 $O(N_C \cdot m \cdot n^2)$ (m 为抗体个数)。本文算法的时间复杂度最坏为: $T(n) = O(N_C \cdot m \cdot n^2) + O(N_C \cdot m \cdot n^2 \log(n)) + O(N_C \cdot m \cdot n^2)$ 。

5.4.5 BAIA 算法测试实验及分析

实验仿真环境: Windows XP 系统, AMD 处理器, 1.90GHZ, 448MB 内存, 仿真软件 MATLAB 7.0。

为了更好地说明 BAIA 算法的有效性, 我们选用国际上通用的 TSPLIB 测试库中的多个实例进行测试。每个测试实例均测试 10 次, 总迭代次数为城市数的 3.5 倍。表中的 Best、Worst、Avg、Error, N_{opt} 分别代表一次迭代找到的最优值、最差值、均值、平均误差, 及得到最优解的次数。

其中

$$\text{Error} = \sum_{i=1}^{10} (S_{T_i} - S_0) / 10 S_0 \times 100\%$$

1. BAIA 算法的参数研究

由于参数 Q 对大多数蚁群算法的影响不大, 在实验中 Q 取 50, 蚁群数量为城市个数, $\alpha = 1$, $\beta = 5$, 克隆算子比例系数 $\alpha = 300$, 克隆算子常数 $b = 20$, 变异概率为 0.3, 总迭代次数为城市数的 3.5 倍, $q_0(t_0) = 0.3, q_{0\min} = 0.1$, 为了对两种状态蚁群比例 ($r_1:r_2$) 进行性能分析, 表 5.5 针对 att48 (城市规模为 48, TSPLIB 提供的已知最优值 $S_0 = 33524$) 进行测试。表 5.6 针对 st70 (城市规模为 70, TSPLIB 提供的已知最优值 $S_0 = 678.5975$) 进行测试。

从表 5.5、表 5.6 可以得出当 $r_1:r_2 = 0.5:0.5$ 时, 算法的性能较好, 最优值、均值、最差值以及误差均优于其他比例。当 $r_1:r_2$ 比值较小时随机性较大, 算法确定性将减弱, 所以算法启发式减弱, 相对来说算法寻优效果较差; 当 $r_1:r_2$ 比值较大时确定性搜索较大, 随机性较小, 算法解空间不大易陷入局部极值, 全局寻优能力较差; 当 $r_1:r_2$ 比值均衡时, 算法在搜索与利用之间找到了平衡点, 能够有效地跳出局部极值, 增强全局寻优能力。

表 5.5 参数 $r_1:r_2$ 对 BAIA 算法的影响 (att48)

$r_1:r_2$	Best	Worst	Avg	N_{opt}	Error(%)
0.9:0.1	33523.7085	34265.9517	33768.0258	1	0.73
0.8:0.2	33523.7085	33804.2399	33630.6158	2	0.32
0.7:0.3	33523.7085	33784.0270	33594.3230	3	0.21
0.6:0.4	33523.7085	33872.1609	33666.4722	1	0.42
0.5:0.5	33523.7085	33699.3207	33586.3621	3	0.19
0.4:0.6	33523.7085	33894.4590	33644.7410	1	0.36
0.3:0.7	33523.7085	33966.1405	33673.7171	2	0.45
0.2:0.8	33523.7085	33784.0270	33634.0727	1	0.33
0.1:0.9	33523.7085	33981.7140	33688.0482	2	0.49

表 5.6 参数 $r_1:r_2$ 对 BAIA 算法的影响 (st70)

$r_1:r_2$	Best	Worst	Avg	N_{opt}	Error(%)
0.9:0.1	677.1096	687.0478	682.5003	1	0.58
0.8:0.2	677.1096	690.2073	683.9836	1	0.79
0.7:0.3	677.1096	683.2623	680.5262	4	0.28
0.6:0.4	677.1096	687.8319	684.5077	1	0.87
0.5:0.5	677.1096	687.5407	679.6102	7	0.15
0.4:0.6	677.1096	687.5407	683.6664	1	0.75
0.3:0.7	677.1096	682.6627	680.4060	1	0.27
0.2:0.8	677.1096	688.6952	682.4554	3	0.57
0.1:0.9	677.1096	688.8032	683.5414	1	0.73

2. BAIA 与相关算法对比实验研究

下面利用 BAIA 的最优参数,通过不同规模的 TSP 问题,比较 BAIA 算法与 ACS、NGA 及 HBACA 算法的性能。选取 $r_1:r_2=0.5:0.5$,其他的参数设置与上述类同。图 5.4 为三种算法求解 st70 的 TSP 问题的收敛曲线。显然,对于 NGA、ACS 算法,随着迭代的进行,算法的确进入了早熟停滞状态,而本文提出的 BAIA 算法则具有收敛快、不进入早熟停滞状态从而搜索出的路径更优(迭代 48 次就收敛了,并且找到最优值 677.1096)。并且,在第一次迭代求的值 BAIA 明显优于 ACS 和 NGA。从表 5.7 中可以看出本文 BAIA 算法,在测试实例 Eil51、Eil76、KroA100、D198 中求得的最差值均优于 ACS,

其中, Eil76、KroA100、D198 中求得的最优值、均值及误差均优于 ACS。这是因为本文将蚁群进行动态分群, 群体之间进行分工合作。将优秀的蚂蚁看做一个个免疫体, 对精英蚂蚁进行克隆选择操作, 促使群体不断进化。同时选取优秀蚂蚁进记忆库, 对记忆库中个体进行局部最优免疫操作, 记忆库中的抗体亲和度不断成熟。所以算法收敛速度、算法多样性保持及求解精度较经典蚁群算法好。

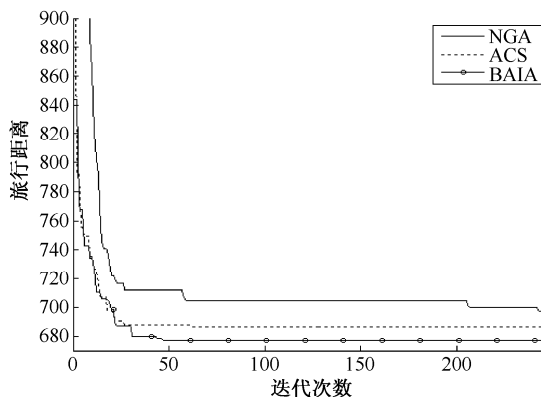


图 5.4 st70 收敛性

图 5.4 中 NGA 为基于 OX 交叉算子且带贪婪算子的遗传算法,

表 5.7 BAIA 算法与 ACS 的比较

实例(S_0)	Method	Worst	Best	Avg	Error(%)
Eil51 (428.87)	ACS	437.80	428.87	429.48	0.14
	BAIA	435.75	428.87	430.79	0.45
Eil76 (544.37)	ACS	565.16	545.97	553.77	1.73
	BAIA	552.80	544.37	548.83	0.82
KroA100 (21285.44)	ACS	22499.74	21285.44	21456.98	0.81
	BAIA	21636.18	21285.44	21359.60	0.35
D198 (15808.65)	ACS	16936.53	15919.55	16143.66	2.12
	BAIA	15924.05	15830.37	15861.29	0.33

5.5 本章小结

本章将人工免疫系统与蚁群算法进行融合,研究了两类免疫蚁群自适应融合算法。第一类,研究了基于统计学习的免疫蚁群自适应融合算法,构造了局部最优搜索算子,充分利用了克隆选择算法(CSA)与蚁群算法(ACO)各自的优势。设计了自适应统计学习控制参数实现了克隆选择、蚁群优化、局部最优搜索策略的有机结合,该算法有效克服了抗体种群“早熟”问题,提高了算法对组合优化问题的求解精度。实验结果证实:ACALA 融合了 ACO-LS 和 CSA-LS 两者的优点,形成优势互补,收敛速度更快,收敛精度更高,其优化性能强于原有基础算法。同时由于蚁群算法最初比较适合于 TSP 组合优化问题求解,因此 ACALA 算法对离散组合优化问题求解具有明显的优势。同时,利用蚁群算法和免疫算法各自特性,将蚁群算法与免疫算法有机结合,还研究了一种双态免疫优势蚁群克隆选择算法。BAIA 算法将蚂蚁分成两种状态,更好地模拟真实蚁群社会中的所有蚂蚁各司其职,又相互依赖协作,形成一个有机整体的客观事实,扩大了解的搜索空间,有效抑制了收敛过程中的早熟停滞现象。ACALA 与 BAIA 这两类算法的实现为人工免疫算法与蚁群算法的融合提供了新的思路,同时也为人工免疫系统与其他智能优化算法(如遗传算法、粒子群算法、人工鱼群算法、差分进化算法等)的自适应融合提供了新的思路,将加快人工免疫系统与其他智能算法的有机结合研究。

第 6 章

基于混合免疫智能算法的 混沌系统自抗扰优化控制

前几章对混合免疫智能算法进行了研究，将研究的混合免疫智能算法应用于非线性混沌系统自抗扰控制中，利用第 3 章研究的免疫双态粒子群算法对高维、多模态问题具有强全局搜索能力、快速实时性等特征，对复杂系统优化控制具有优势。本章将研究两种混沌系统自抗扰优化控制方法，第一，研究一种基于免疫双态粒子群算法的混沌系统自抗扰优化控制器；第二，研究一种基于免疫双态粒子群算法的自抗扰神经网络并行优化控制方法。并通过实验验证本章所提的混沌系统优化控制方法是有效的，能够实现非线性混沌系统同步跟踪与抗干扰控制。

6.1 引言

混沌是非线性动力学所特有的一种运动形式，对初始值极为敏感，其行为运动轨迹非常复杂。近年来混沌系统在数据加密^[95~97]、保密通信^[98~100]、混沌电路设计^[101,102]、生命医学^[103]等诸多领域有着广阔的应用前景。同时现代工业过程中也包含相当一部分非线性离散混沌系统模型，如机器人操作手转移系统^[173]、重油热解炼制工业^[83]、发动机引擎控制系统^[174]、六辊 UC 轧机系统^[175]等领域，因此研究离散混沌系统控制也显得尤为重要。过去人们一直认为混沌是不可控制的，但是自从 Ott 等人^[104]提出了基于参数扰动的方

法,成功地实现了混沌系统的控制。自此混沌系统的控制问题逐渐引起了人们浓厚的兴趣,许多研究人员进行了一些有意义的探索,新的研究成果不断出现。现有的控制方法主要有神经网络控制^[105]、自适应控制^[106,107]、模糊控制^[108,109]、智能 PID 控制^[110]等,提高了混沌系统的控制效果,但现有方法大多需要系统具有精确数学模型。然而实际物理系统内部参数的摄动及外部的扰动将给混沌系统带来诸多不确定性,难以确定系统的精确模型,给混沌系统控制带来了较大的难度^[111]。

自抗扰控制器(ADRC)继承了 PID 的优点,是一种不依赖于精确系统模型的非线性鲁棒控制方法,具有控制精度高、响应速度快、抗干扰能力强等特点^[112],对摄动与外部扰动具有良好的鲁棒性,近年来被广泛地应用于非线性系统控制中^[113]。文献[176]提出了一种混沌系统的扩张观测器(ESO),自抗扰控制技术开始应用于非线性混沌系统控制中,但这方面研究成果还不多见。相关研究表明自抗扰控制器能否充分发挥其优越性,关键在于实际应用中控制器参数能否得到最优配置。ADRC 需要整定的参数较多,整定的过程和效果以往在很大程度上依赖于人的经验,不可避免会遇到局部极值问题。因此,参数整定问题已成为目前 ADRC 研究的焦点之一。对自抗扰控制器的参数优化研究现有成果主要有基于时间尺度优化方法^[177]、粒子群优化方法^[178]、遗传算法、自适应人工免疫优化方法^[179]等,并取得了一定的成果。但面对复杂系统控制如何获得一组较优 ADRC 的参数,使整个控制系统具有更好的控制品质,将是自抗扰控制技术研究的一大热点。

自抗扰控制器对实时动态系统控制有所欠缺,而小脑模型神经网络^[180](Cerebellar Model Articulation Controller, CMAC),是仿照小脑控制肢体运动原理而建立的神经网络模型,具有很强的局部非线性逼近能力和一定的泛化能力,其参数较少,实现简单,学习速度快,广泛应用于非线性系统控制等领域^[181],且能够克服 ADRC 在控制非线性系统时动态性能差的缺点。但神经网络控制的稳定性受到迭代初值和训练程度的影响,控制初期效果不佳,且抗干扰能力不强,将 ADRC 与 CMAC 进行结合能很好地融合二者的优点从而构成并行控制器。

在第 3 章中,研究了一种免疫双态粒子群算法,该算法具有快速收敛性,全局寻优能力强,且对高维及多模态问题的求解具有优势,适合于复杂系统控制器优化设计。混沌系统自抗扰优化控制系统同样具有多模态优化问题的

1. 跟踪微分器

跟踪微分器一个动态环节, 对其输入信号, 它将给出两个输出, 其中一个跟踪输入信号, 另一个是输入信号的导数, 可以有效抑制信号噪声干扰。

$$\begin{cases} x_1(k+1) = x_1(k) + T \cdot x_2(k) \\ x_2(k+1) = x_2(k) + T \cdot \text{fst}[x_1(k), x_2(k), r, h_0] \end{cases} \quad (6.1)$$

式 (6.1) 中, T 为采样步长, r 为决定跟踪快慢的参数, h_0 为滤波因子, 非线性函数 fst 如式 (6.2) 所示。

$$\text{fst} = - \begin{cases} ra/d, & |a| \leq d \\ r \text{sgn}(a), & |a| > d \end{cases} \quad (6.2)$$

式 (6.2) 中的参数变量关系如下:

$$\begin{cases} d = rh_0 \\ d_0 = dh_0 \\ y(k) = x_1(k) - y_0 + h_0 x_2(k) \\ a_0 = \sqrt{d^2 + 8r|y(k)|} \\ a = \begin{cases} x_2 + \text{sign}[y(k)](a_0 - d)/2, & |y| > d_0 \\ x_2 + y(k)/h_0, & |y| \leq d_0 \end{cases} \end{cases} \quad (6.3)$$

2. 扩张状态观测器

扩张状态观测器是自抗扰控制器的核心部分。运用非线性结构对系统的状态和模型的不确定及外部扰动进行估计。

$$\begin{cases} e(k+1) = z_1(k) - y(k+1) \\ z_1(k+1) = z_2(k) + T \cdot [z_2(k) - \beta_{01}e(k)] \\ z_2(k+1) = z_2(k) + T \cdot [z_3(k) - \beta_{02}\text{fal}(e, 0.5, \delta) + b_0 u(k)] \\ z_3(k+1) = z_3(k) - T \cdot \beta_{03}\text{fal}(e, 0.25, \delta) \end{cases} \quad (6.4)$$

式 (6.4) 中, β_{01} 、 β_{02} 、 β_{03} 为输出误差校正增益可调参数, 非线性函数 $\text{fal}()$ 如下:

$$\text{fal}(e, \alpha, \delta) = \begin{cases} |e|^\alpha \text{sgn}(e), & |e| > \delta \\ e/\delta^{1-\alpha}, & |e| \leq \delta \end{cases}, \quad \delta > 0 \quad (6.5)$$

3. 非线性状态误差反馈律

在非线性状态误差反馈控制律中, 通过将传统的 PID “线性和” 形式转换为 “非线性组合”, 从而得到非线性 PID, 以此提高系统的控制品质。

$$\begin{cases} e_1(k+1) = x_1(k+1) - z_1(k+1) \\ e_2(k+1) = x_2(k+1) - z_2(k+1) \\ u_0(k+1) = \beta_1 \text{fal}[e_1(k+1), \alpha_1, \delta] + \beta_2 \text{fal}[e_2(k+1), \alpha_2, \delta] \\ u(k+1) = u_0(k+1) - z_3(k+1)/b_0 \end{cases} \quad (6.6)$$

其中 β_1, β_2, b_0 为可调参数。自抗扰控制器只需要对象的输入 y_0 , 输出 y_{out} 和控制器的目标值, 不依靠受控对象的精确模型。因此这种控制方案在工程实际中容易实现。由上述可知, 对控制器参数除了 $\{\beta_{01}, \beta_{02}, \beta_{03}, \beta_1, \beta_2, b_0\}$ 需要调试外, 其他控制器参数都可以设置成固定参数。引入智能优化理论, 对以上参数进行在线实时优化, 于是在这里采用第3章研究的免疫双态粒子群算法对 ADRC 的以上参数进行在线优化, 促使 ADRC 控制器具有自学习、自适应能力。

6.3 基于免疫双态粒子群算法的混沌系统自抗扰优化控制

6.3.1 非线性混沌系统

考虑如下离散非线性混沌系统:

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-m), u(k), u(k-1), \dots, u(k-n)] + f_N(k) \quad (6.7)$$

其中 $y(k), u(k) \in R$ 分别为 k 时刻的系统输出、输入信号, $f_N()$ 表示未知动态、扰动或其他参数不确定性等因素。 $m, n \in R$ 为系统输出, 输入阶数, $f()$ 是未知非线性函数。

6.3.2 评价函数的选择

控制系统最优指标的选取不仅考虑系统的快速性、稳定性及准确性, 同时也应考虑控制能量问题, 其形式为

$$J(p)=\int_0^{\infty}(w_1|e(t)|+w_2|u(t)|)dt \tag{6.8}$$

式（6.8）中 $e(t)$ 为系统误差， $u(t)$ 为控制量， w_1, w_2 为权重，适应度函数 $J(p)$ 值越小，表明相应粒子 p 越靠近全局最优解。评价函数随着算法的运行逐步减小。

6.3.3 混沌系统自抗扰优化控制系统结构图

其系统结构图如图 6.2 所示。

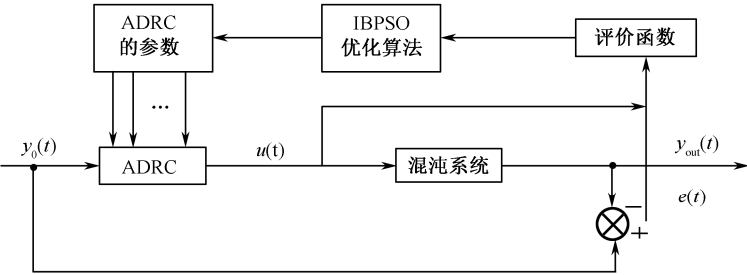


图 6.2 基于 IBPSO 算法混沌系统自抗扰控制系统框图

6.3.4 基于 IBPSO 算法的混沌系统 ADRC 优化控制算法流程

基于 IBPSO 算法的混沌系统 ADRC 优化控制算法流程描述

begin

步骤 1: ADRC 控制器等参数初始化

步骤 2: while（不满足退出条件）do //（退出条件为设定的截止代数）

for 对粒子群中每只粒子 do

根据式（3.3）、式（3.5）更新粒子的位置与速度，根据系统评价函数计算各粒子适应度值。

计算粒子进化因子 f 。

if (($f < 1$)) {

```

    if ((u(0,1)<1-f)
then
    将粒子群的适应值排序，性能较好的 25%按捕食状态行动，
    其中捕食状态中的部分较优粒子——精英粒子进行增强学习；
    其余粒子按探索状态行动，探索种群按照式 (3.7)、式 (3.8) 更新粒子的位置与
    速度。}
end if
    如果各粒子适应度优于相应粒子历史最优适应度，相应粒子  $P_{id}$  更新。
    如果最优粒子适应度优于历史最优粒子适应度， $P_{gd}$  更新。
end for
    步骤 3: 选择  $k$  ( $k < N$ ,  $N$  为粒子个数) 个最好的抗体个体组成一个临时的克隆子种
    群  $sub_1$ ，对子群体  $sub_1$  进行克隆操作，克隆变异，免疫选择。
    步骤 4: 对趋于凋亡的微粒进行受体编辑。
    步骤 5: 控制器输出，对混沌系统进行控制。
end while
    步骤 6: 系统输出，算法运行结束。
end

```

6.3.5 实验

1. 算例仿真

为了证明 IBPSO-ADRC 的有效性，采用一个被控对象为时变非线性系统进行仿真，被控对象模型^[182]如下。

实例 1:

$$y_{out}(k) = \frac{a(k)y_{out}(k-1)}{1 + y_{out}^2(k-1)} + u(k-1) \quad (6.9)$$

式 (6.9) 中，系数 $a(k)$ 是时变的， $a(k) = 1.2(1 - 0.8e^{-0.1k})$ 。其阶跃跟踪设定值为 $r=1.0$ 。

由图 6.3 可以看出，基于免疫双态粒子群算法优化 ADRC 控制系统 (IBPSO-ADRC)，其响应速度、调节时间、超调量、误差等性能方面明显优

于常规的 PSO 算法优化的 ADRC 控制^[178]、BP 网络优化的 PID 控制^[182] (BP-PID)、基于模糊 RBF 网络优化的 PID 控制 (Fuzzy-RBF-PID)^[182]。这是由于免疫双态粒子群超强快速全局搜索能力, 快速地找到了 ADRC 控制器参数全局最优值。

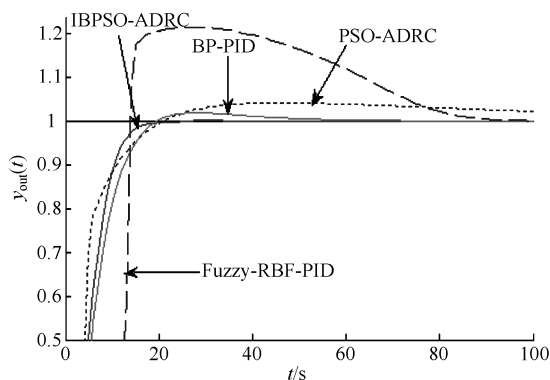


图 6.3 仿真结果比较

2. 混沌系统仿真实验

(1) 确定性混沌系统

带控制项的 Henon 混沌系统^[110,183]方程为

$$\begin{cases} x_1(k+1) = 1 - a(x_1(k))^2 + bx_2(k) + u(k) \\ x_2(k+1) = x_1(k) \end{cases} \quad (6.10)$$

选择系统参数 $a=1.4, b=0.3$, 当 $u(k)=0$ 时系统处于混沌态。其不稳定平衡点为 $(0.6314, 0.6314)$, 在 $t=30s$ (第 30 步) 加入控制量, 经过 IBPSO 算法数代优化后的 ADRC 参数 $\{\beta_{01}, \beta_{02}, \beta_{03}, \beta_1, \beta_2, b_0\} = \{-846.75, -50762.97, -1480798.40, -220.69, 7.81, -585142468.35\}$ 。系统状态变化曲线、控制量、误差曲线及优化评价函数变化曲线如图 6.4 所示。从图 6.4 可以得出, 在 30 步之前没对系统加以控制, 系统处于混沌状态, 一旦加了控制量以后, 系统能在很短的几步时间内达到平衡态。

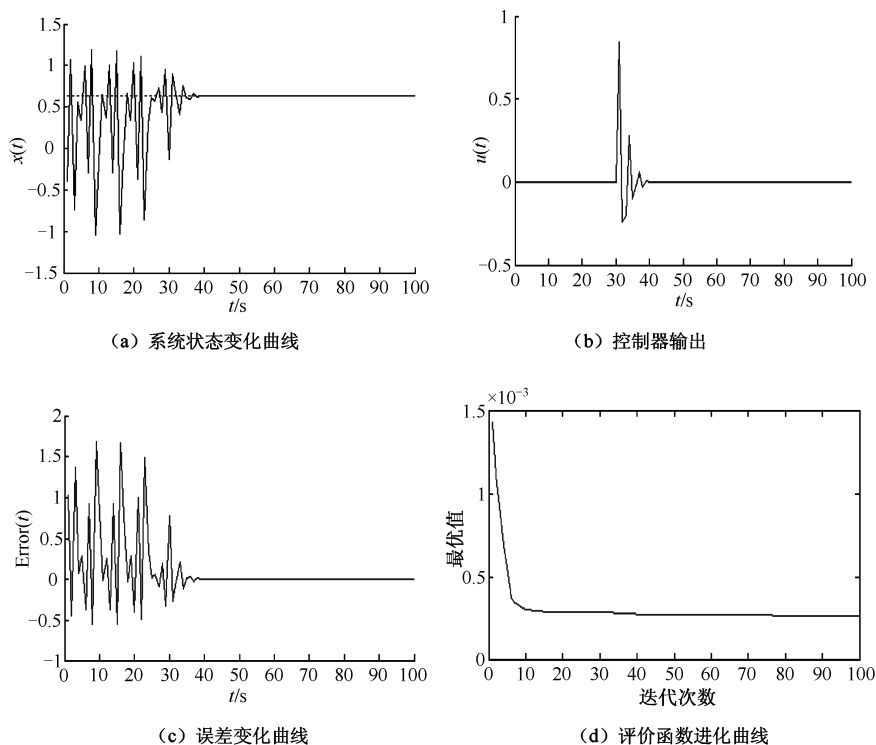


图 6.4 确定性 Henon 混沌系统的 ADRC 控制结果

(2) 混沌系统正弦信号追踪

给定输入信号 $y_0(t)=0.25\sin(t/2)$, 在 $t=30\text{s}$ 时加入控制量, 经过 IBPSO 算法优化的 ADRC 参数 $\{\beta_{01}, \beta_{02}, \beta_{03}, \beta_1, \beta_2, b_0\}=\{834.47, 448812.24, -29812122.56, -215.00, -13.15, 32453049.84\}$ 。数值仿真结果如图 6.5 所示, 从图 6.5 中可以看出, 系统在加入控制量后跟踪的过程中不再出现混沌现象, 相轨迹跟踪了正弦周期轨道, 有较好的跟踪效果。

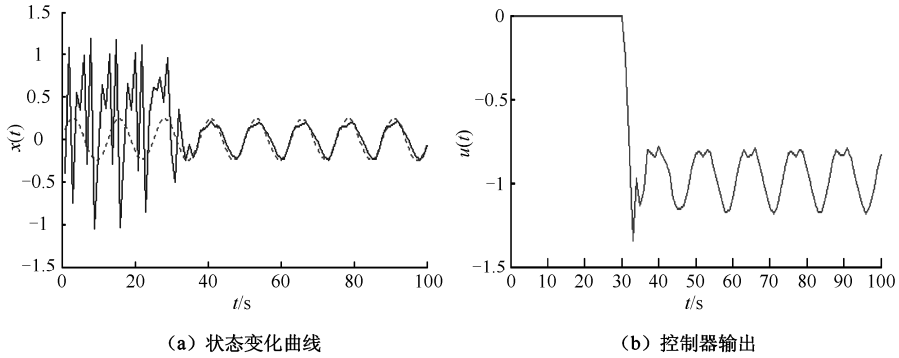


图 6.5 Henon 混沌系统的正弦跟踪 ADRC 控制结果

(3) 不确定性混沌系统

Henon 混沌系统方程^[110,183]为

$$\begin{cases} x_1(k+1) = 1 - (a + \sigma a)(x_1(k))^2 + (b + \sigma b)x_2(k) + \sigma d_1 + u(k) \\ x_2(k+1) = x_2(k) + \sigma d_2 \end{cases} \quad (6.11)$$

其中不确定性为 $\sigma a = 0.02$, $\sigma b = 0.03$, 外部干扰为 $\sigma d_1 = 0.1\sin(k)$, $\sigma d_2 = 0.01\sin(k)$ 。在 $t=30\text{s}$ (第 30 步) 时对系统加入控制量, 经过 IBPSO 算法优化的 ADRC 参数 $\{\beta_{01}, \beta_{02}, \beta_{03}, \beta_1, \beta_2, b_0\} = \{1422.77, 1136347.07, 1386587.33, 87, 15.29, -71622831.42\}$ 。系统状态变化曲线、控制量、误差曲线、优化适应度变化曲线如图 6.6 所示。系统在内部参数摄动及外部干扰的情况下能在很短的时间内达到平衡态, 可以看出文中的方法对不确定性混沌系统具有很好的控制效果。通过不确定性混沌系统控制表明, IBPSO-ADRC 控制器兼顾了 ADRC 的抗干扰优势及 IBPSO 的参数全局寻优特性。系统的控制效果如图 6.6 所示。

(4) 六辊 UC 轧机混沌系统应用实例

六辊 UC 轧机为典型的混沌系统^[175]模型如下:

$$x(n+1) = r \cdot x(n) \exp(1 - x(n)^2) + u(n) \quad (6.12)$$

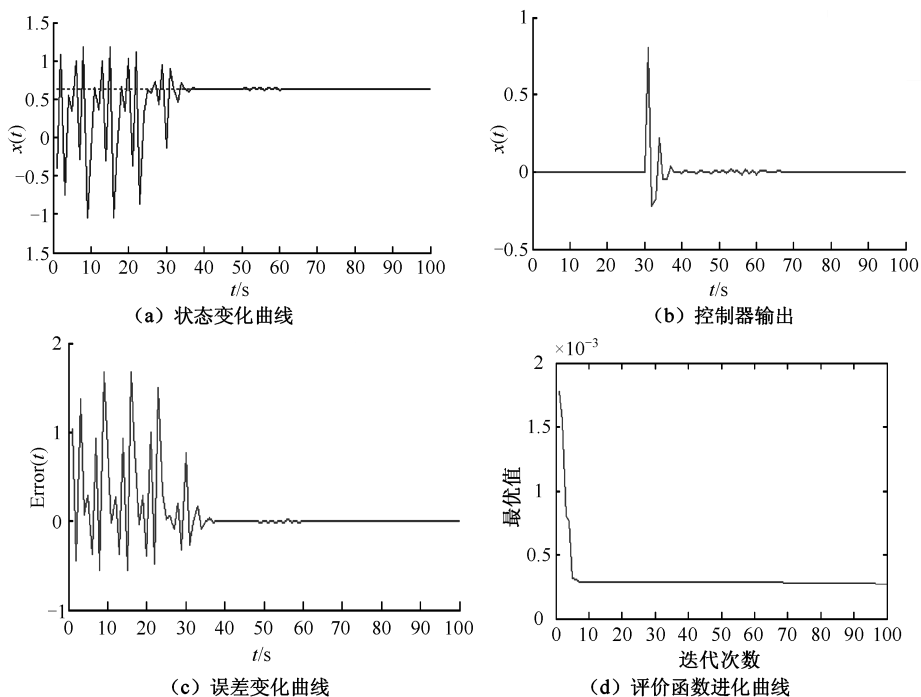


图 6.6 非确定性 Henon 混沌系统的 ADRC 控制结果

未给系统施加控制作用时,即 $u(n)=0$ 时,在 $r \geq 1.5$ 情况下,系统处于混沌状态。初始条件 $x(0)=1.45$,当 $t \geq 50$ 时加入控制量,目标是控制 $x(n)$ 到系统平衡点 1.199273。经过 IBPSO 算法优化后的 ADRC 参数 $\{\beta_{01}, \beta_{02}, \beta_{03}, \beta_1, \beta_2, b_0\} = \{-132.73, 388046.48, 1156883.80, 195.61, 15.47, 242206231.95\}$ 。六辊 UC 轧机混沌系统的控制效果如图 6.7 所示。

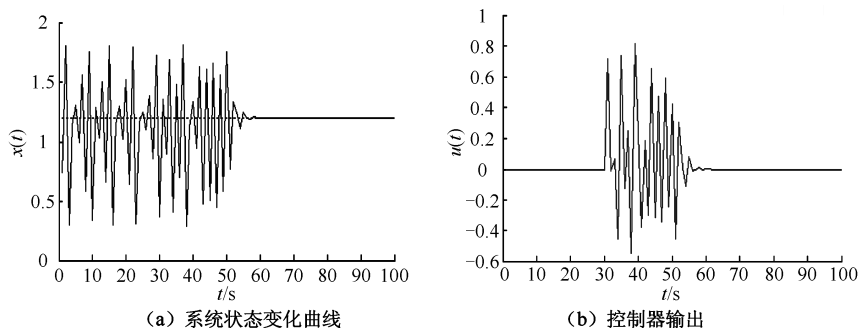


图 6.7 六辊 UC 轧机混沌系统的 ADRC 控制结果

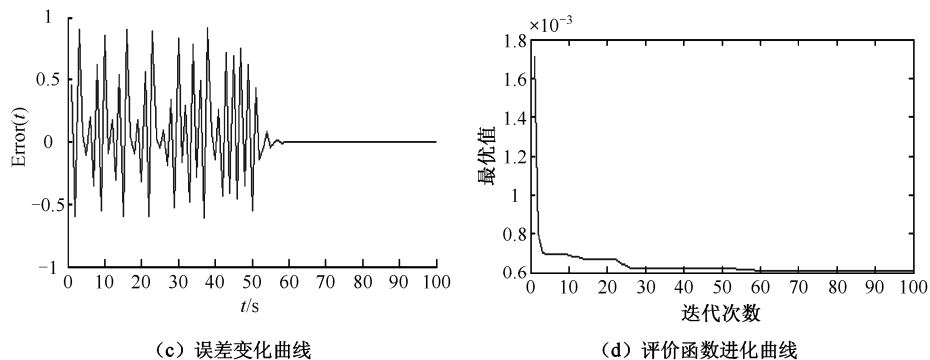


图 6.7 六辊 UC 轧机混沌系统的 ADRC 控制结果 (续)

6.4 基于 IBPSO 算法的混沌系统 ADRC-CMAC 并行优化控制

6.4.1 CMAC 结构

CMAC 网络^[180,181]由输入层、中间层和输出层组成。CMAC 的输入输出非线性关系由概念映射 ($U \rightarrow AC$) 和实际映射 ($AC \rightarrow AP$) 来实现, 其结构如图 6.8 所示。

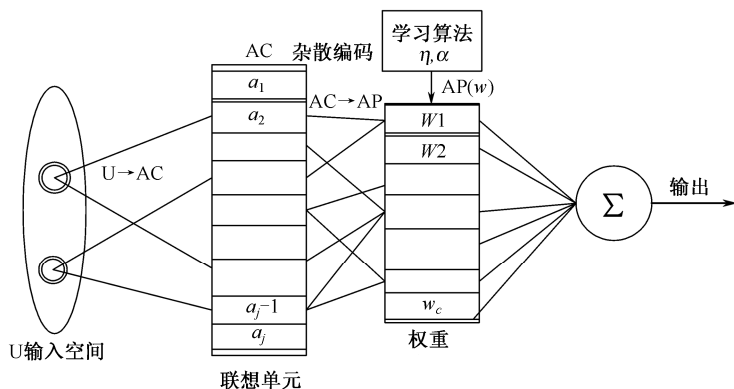


图 6.8 CMAC 结构图

CMAC 与其他神经网络相比具有明显的优越性：如它是基于局部映射的神经网络，每次修正的权值较少，学习速度快，实时能力强；具有一定的泛化能力；具有连续输入、输出能力等特征。该神经网络非线性逼近能力强，适合于复杂动态环境下的非线性系统实时控制。图 6.8 中 η 为网络学习率， α 为惯性常量且 $\alpha \in (0,1)$ ， η 、 α 参数的合适选取对 CMAC 神经网络的运行性能有影响。

6.4.2 ADRC 与 CMAC 并行控制算法

ADRC-CMAC 并行控制系统由 ADRC 和 CMAC 神经网络复合构成。ADRC 实现反馈控制，根据系统的输出误差，产生反馈信号，保证控制过程中系统的稳定性，且抑制系统的扰动；CMAC 通过学习实现被控对象的逆动态模型，在控制初期对 ADRC 进行学习，对系统进行实时动态跟踪；免疫双态粒子群算法 (IBPSO) 对 ADRC-CMAC 控制器参数进行全局优化，以保证 ADRC-CMAC 控制器参数得到最优配置 (图 6.9)。

系统的控制算法为

$$u_{\text{cmac}}(k) = \sum_{i=1}^c w_i a_i \quad (6.13)$$

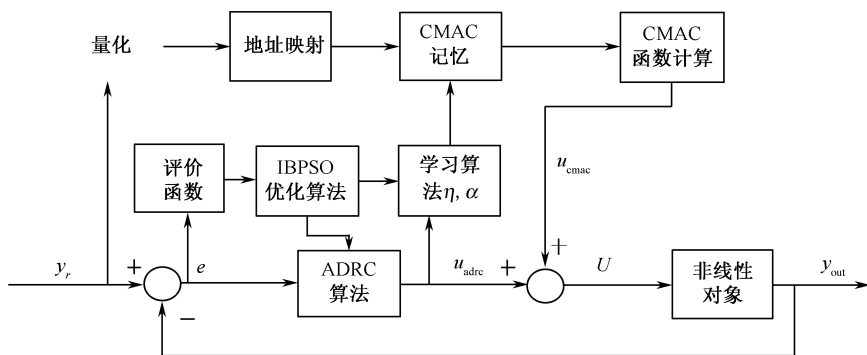


图 6.9 ADRC 与 CMAC 并行优化控制结构图

$$U(k) = u_{\text{adrc}}(k) + u_{\text{cmac}}(k) \quad (6.14)$$

式 (6.13) 中， a_i 为二进制数据选择向量， c 为 CMAC 网络的泛化参

数；式 (6.14) 中 $u_{\text{cmac}}(k)$ 为 CMAC 产生的相应的输出， $u_{\text{adrc}}(k)$ 为 ADRC 控制器产生的输出， $U(k)$ 为总控制器输出。

CMAC 采用有导师的学习算法，每一控制结束时，计算出相应的 CMAC 输出 $u_{\text{cmac}}(k)$ ，并与总控制器输入 $u(k)$ 相比较，修正权重，进入学习过程。

CMAC 的调整目标为

$$E(k) = \frac{1}{2} (u_n(k) - u(k))^2 \cdot \frac{1}{c} \quad (6.15)$$

$$\Delta w(k) = -\eta \frac{\partial E(k)}{\partial w} = \eta \frac{u(k) - u_n(k)}{c} a_i = \eta \frac{u_a(k)}{c} a_i \quad (6.16)$$

$$w(k) = w(k-1) + \Delta w(k) + \alpha(w(k) - w(k-1)) \quad (6.17)$$

当系统开始运行时，系统由 ADRC 控制器进行控制，保证系统的稳定、抗干扰，通过 CMAC 的不断学习逐渐快速逼近实际系统。随着系统误差信号逐渐减小，使 ADRC 产生的输出控制量 $u_{\text{adrc}}(k)$ 逐渐减弱，直到 CMAC 产生的输出控制量 $u_{\text{cmac}}(k)$ 逐渐逼近控制器。CMAC 控制算法虽然是由 ADRC 控制器的输出训练学习而来的，但并不是 ADRC 控制器的简单复制。ADRC 控制器是为了评判 CMAC 控制器的性能，增强系统的稳定性、抑制系统扰动。利用免疫双态粒子群算法 (IBPSO) 对 ADRC-CMAC 控制器参数集合 $\{\beta_{01}, \beta_{02}, \beta_{03}, \beta_1, \beta_2, b_0, \eta, \alpha\}$ 进行寻优可以克服普通 ADRC 控制器没有学习能力以及控制参数难以制定的弊端，使控制效果得到提高。

6.4.3 基于 IBPSO 的 ADRC-CMAC 优化设计流程

其流程如图 6.10 所示。

6.4.4 实验

1. 数值仿真实例

为了验证文中控制算法的强抗干扰能力，选取一类时刻强正弦扰动离散非线性系统进行测试，并与最近算法进行比较，其系统模型来自文献[184]。

$$y(k+1)=\frac{y(k)}{1+y(k)^2+d(k)}+u(k)^3 \tag{6.18}$$

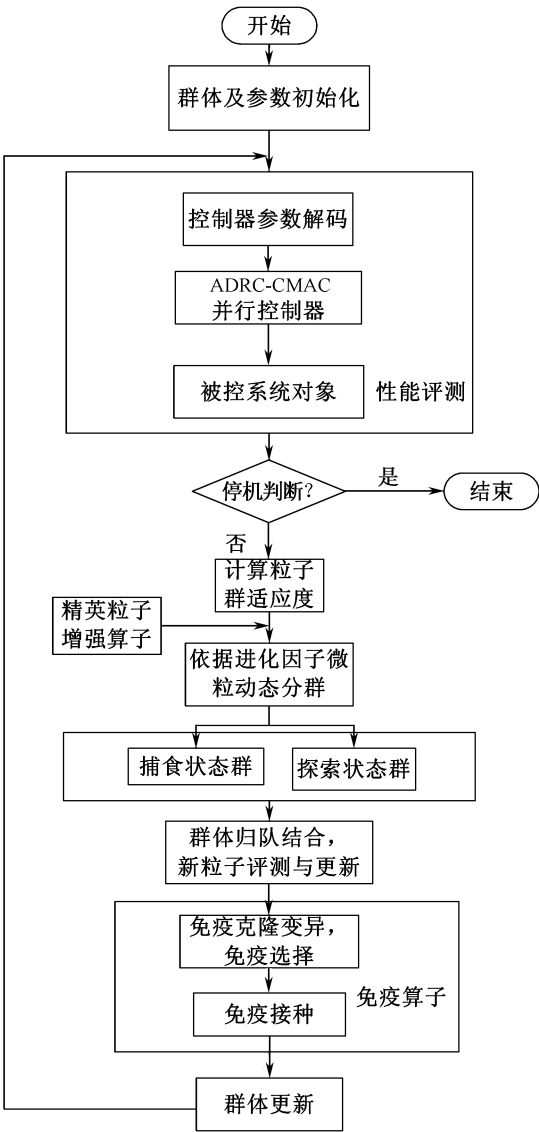


图 6.10 基于免疫动态粒子群的 ADRC-CMAC 并行优化控制流程

期望信号: $y_r(k+1)=1$

扰动信号: $d(k)=\sin(k/50)$

表 6.1 中 NN-SMC 为一种基于神经网络估计的非线性离散时间系统自适应准滑模控制方法^[184]。NN-SMC1 为基于紧格式、NN-SMC2 为基于偏格式、NN-SMC3 为基于全格式线性化得出的 NN-SMC 控制律。BP-PID 为基于 BP 网络整定的 PID 控制算法^[182]。 $\delta\%$ 表示系统超调量, $J=\int_0^t |y_r-y_{\text{out}}|dt$ 为绝对误差的一阶积分。

表 6.1 五种算法性能指标比较

P	NN-SMC1	NN-SMC2	NN-SMC3	BP-PID	ADRC-CMAC
$\delta\%$	2.27%	2.13%	2.06%	4.26%	1.96%
J	6.67	4.65	4.57	11.54	4.00

通过图 6.11 和表 6.1 可以清楚地看到本书 ADRC-CMAC 方法在抗扰动、超调量及误差绝对值等各项性能指标上均优于 NN-SMC1、NN-SMC2、NN-SMC3 及 BP-PID 方法。

2. 不确定 Henon 混沌系统控制

Henon 混沌系统方程^[110,183]为

$$\begin{cases} x_1(k+1)=1-(a+\sigma a)(x_1(k))^2+(b+\sigma b)x_2(k)+\sigma d_1+u(k) \\ x_2(k+1)=x_2(k)+\sigma d_2 \end{cases} \quad (6.19)$$

当 $u(k)=0$ 时系统处于混沌态。其中不确定性为 $\sigma a=0.02$, $\sigma b=0.03$, 外部干扰为 $\sigma d_1=0.1\sin(k)$, $\sigma d_2=0.01\sin(k)$, 其不稳定平衡点为(0.6314,0.6314), 为了验证本书控制算法的有效性, 在 $n=30$ (第 30 步) 时对系统加入控制量, 控制到平衡态。从图 6.12 (a) 可以看出混沌系统在内部参数摄动与外部扰动的情况下, 本书 ADRC-CMAC 控制方法很快控制到平衡点, 具有很强的抗干扰能力; 由图 6.12 (b) 可知控制初期主要由 ADRC 控制器起作用, 后期作用减弱且有微微的振动, 但一直起到抑制系统干扰作用, CMAC 不断学习迅速逼近系统, 动态性能较好; 图 6.12 (d) 所示控制适应度曲线收敛速

度较快,说明文中提出的免疫动态粒子群算法全局寻优能力强。

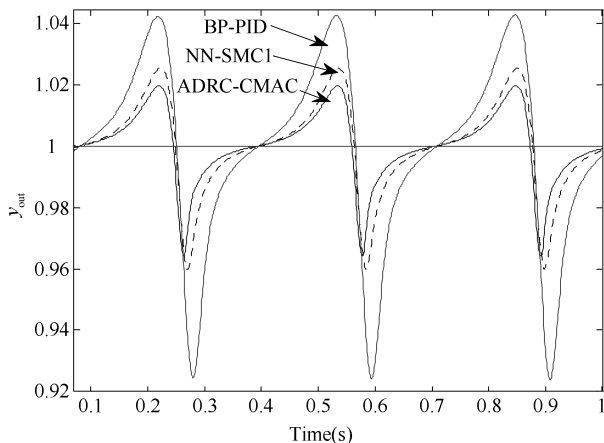


图 6.11 三种方法的比较

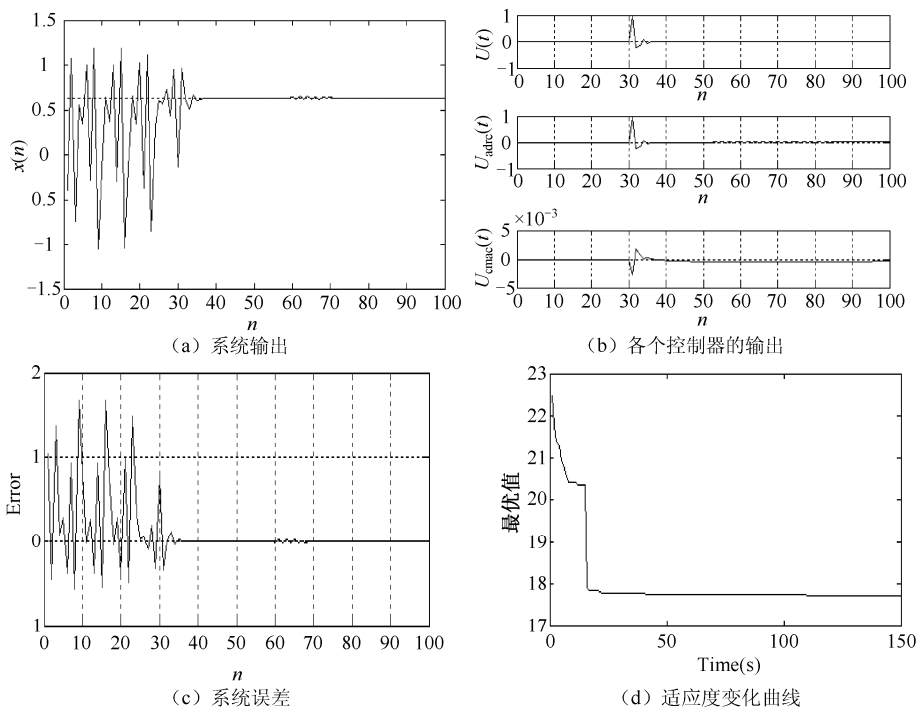


图 6.12 不确定 Henon 混沌系统控制

3. 六辊 UC 轧机混沌系统控制

(1) 无噪声环境下的轧机混沌系统控制

六辊 UC 轧机混沌系统^[175]模型为

$$x(n+1) = rx(n)\exp[1-x(n)^2] + u(n) \quad (6.20)$$

此系统未施加控制作用, 当 $U(n)=0$ 时, 在 $r \geq 1.5$ 的情况下, 系统处于混沌态。初始条件为 $x(0)=1.45$, 系统平衡点为 1.199273。而工程实际中所要求的控制目标往往不在系统的平衡点, 能够控制系统到任一非平衡点这是混沌系统控制的一大难题, 为了验证算法的有效性, 当 $n \geq 50$ 时加入控制作用, 控制到非平衡态, 取给定为恒值 $x_r(n)=2$, 从图 6.13 可以看出 ADRC-CMAC 控制方法可以控制混沌系统到某一给定恒值。

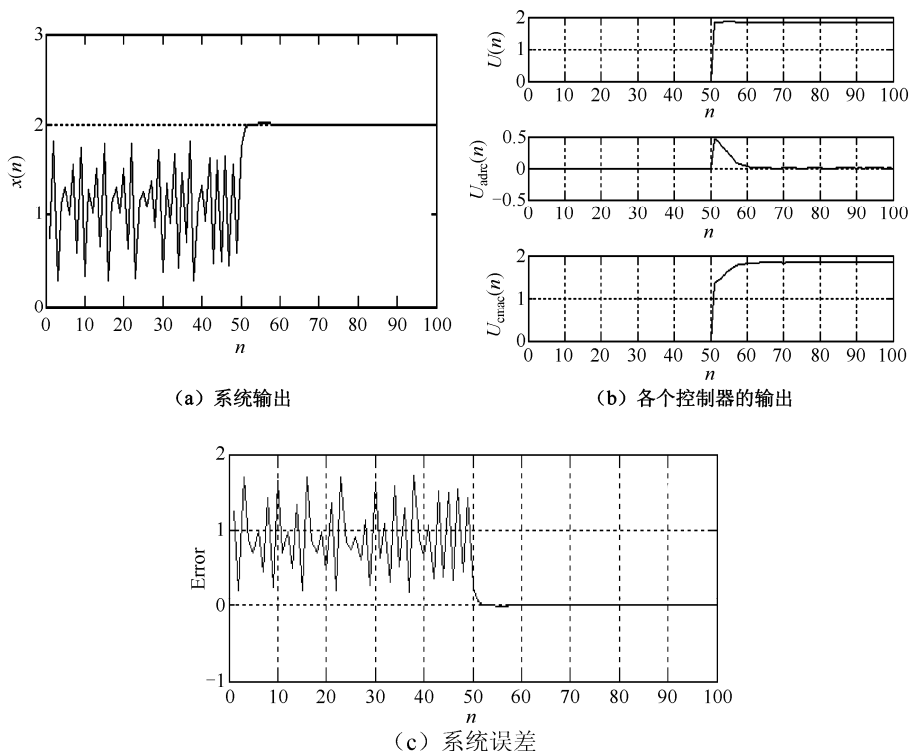


图 6.13 给定值下六辊 UC 轧机混沌系统控制

(2) 噪声环境下的轧机混沌系统控制

带扰动的六辊 UC 轧机混沌系统模型如下:

$$x(n+1) = rx(n)\exp[1-x(n)^2] + u(n) + \xi(k) \quad (6.21)$$

其中 $\xi(k)$ 为均值为 0, 方差为 0.01 的高斯随机噪声。当 $n \geq 50$ 时加入控制作用, 控制到非平衡态, 取给定为恒值 $x_r(n)=2$, 从图 6.14 可以看出 ADRC-CMAC 控制方法在混沌系统外加扰动的情况下仍然能对系统进行有效的控制。

从实验结果可以看出控制初期主要由 ADRC 控制器起作用, 抑制干扰, CMAC 控制器对 ADRC 快速学习, 加快控制响应速度, 后期 CMAC 控制器起主要作用。同时在干扰的环境下, ADRC 能有效地抗干扰, CMAC 控制器快速地实现跟踪。IBPSO 算法对 ADRC-CMAC 参数全局快速寻优, 使得控制效果得到大幅度提高。基于 IBPSO 的 ADRC-CMAC 控制系统兼顾了抗扰动能力、鲁棒性好、学习能力、输出误差小、快速全局收敛等特征, 可为复杂非线性被控对象提供参考。

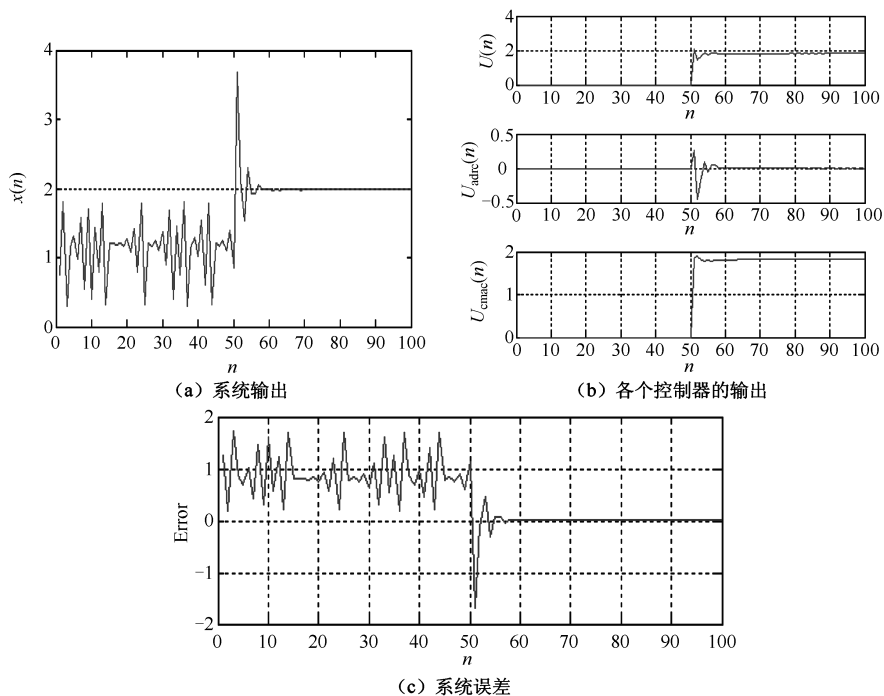


图 6.14 噪声环境下六辊 UC 轧机混沌系统控制

6.5 本章小结

针对混沌系统难以控制, 将自抗扰控制引入混沌系统控制中, 利用文中所研究的免疫双态粒子群优化算法具有强全局搜索能力及快速实时性等优点, 将其用于复杂系统控制器优化设计。将免疫双态粒子群算法应用于混沌系统自抗扰控制器优化设计, 构建了两种混沌系统优化控制方法, 第一种方法为基于免疫双态粒子群算法的混沌系统自抗扰控制系统, 另一种方法为基于免疫双态粒子群的混沌系统自抗扰神经网络并行优化控制方法。利用免疫动态粒子群算法对控制器参数进行快速寻优, 有效避免了寻优过程陷入局部极小问题, 从而实现了系统的优化控制。对一类受扰离散非线性系统、不确定 Henon 混沌系统及六辊 UC 轧机混沌系统进行控制仿真, 结果证明该算法在控制品质上有很大的提高, 在系统动态响应速度、稳定性、抗干扰能力等方面都表现出很好的性能。为离散非线性混沌系统控制提供了一种新的智能控制方法。

第 7 章

基于混合免疫智能算法的永磁同步电机系统多参数辨识与状态监测

实际 PMSM 系统是一类时变、耦合、强非线性动态系统，其系统参数辨识难度大，获得系统准确的物理参数是此类系统建模、优化控制、状态监测与稳定运行的关键。在第 4 章中，研究了一种免疫协同粒子群进化算法，该算法具有大空间快速搜索能力、稳健较好及动态目标优化能力等特性，适合于复杂非线性系统建模与参数辨识。本章将免疫协同粒子群进化算法应用于永磁同步电机多参数辨识中，构建一种永磁同步电机多参数高效智能辨识与温度监测模型方法，实验结果表明该方法是有用的、可行的。

7.1 引言

永磁同步电机（PMSM）具有响应速度快、控制性能好、高密度功率等优点，被广泛应用于工业驱动系统中^[114]。其系统参数易受温度、定子电流、磁通饱和等变化因素影响，造成磁链观测不准确和 d - q 轴电流无法彻底解耦，给控制系统带来很大难度，降低了系统运行的可靠性^[115]。因此 PMSM 电机参数能否准确辨识影响到电机控制性能以及电机状态的准确监测，而永磁同步电机系统为典型的非线性动态欠秩系统，能够对永磁同步电机多参数精确辨识具有重要的理论与实际意义。研究人员进行了一些有意义

的尝试与探索, 现有永磁同步电机参数辨识方法主要有最小二乘法^[185]、基于扩展卡尔曼滤波的方法^[186]、基于神经网络^[125,187]的方法、基于模型参考自适应的方法^[188,189]、快速傅里叶提取基波计算方法^[190]、进化计算方法^[191,192]等。

最小二乘法^[185]与快速傅里叶提取信号计算方法^[190]建立在经典数学理论基础之上, 需要进行大量复杂的数学计算, 一旦电机运行工况改变, 将影响到辨识结果; 基于扩展卡尔曼滤波的辨识方法^[186]可以过滤掉辨识过程中的噪声信号, 能同时辨识电机的状态和参数。但该方法需要对电机数学模型预先处理且需要进行大量的矢量矩阵运算。参考文献[125]分析现有电机欠秩参考模型并设计一种满秩模型, 建立一种基于 Adaline 神经网络的永磁同步电机参数辨识方法, 能有效地辨识电阻、电感、磁链。该方法将电机 d 轴与 q 轴电感等价对待, 实际上电机运转时 d 轴与 q 轴电感不对称, 辨识参数不够准确。参考文献[188]提出了一种模型参考自适应的参数辨识方法, 该方法先设定电机电阻为数据手册上的设计值, 然后再去辨识转子磁链; 以同样的原理设置电机转子磁链为数据手册上的设计值, 然后再去辨识定子电阻, 然而电机在实际工况运行过程中各参变量实际值不完全等于其设计值, 该方法给辨识带来了较大的误差。参考文献[192]提出了一种基于粒子群的 PMSM 参数辨识, 该方法能较好地辨识电阻和转矩, 但不能辨识变化的磁链与电感。由于实际 PMSM 系统的强非线性、目标函数存在多局部极值点等, 常规方法难以得到高精度参数估计值。如何对永磁同步电机多参数准确辨识仍然是一大难题。

同时, 由于电流的热效应, 电机运行过程中会发热, 导致温度升高, 如电机过热运行超出其极限温度范围将导致电机寿命降低并影响电拖动系统安全运行。因此对 PMSM 系统温度在线监测具有重要的意义。近年来, 永磁同步电机在线温度监测研究取得了新进展^[193,194], 现有大部分研究基于温度传感器等辅助设备完成, 而辅助设备的安装与维护代价较大, 限制了其广泛应用。研究一种简便实用的 PMSM 系统定转子温度监测软测量方法将成为 PMSM 系统工况识别的研究热点。从参数辨识方法中得到启发, 基于智能参数辨识技术的电机温度在线监测只需要采集电机的定子电压、定子电流及转速等信息就能辨识出电机稳态运行时定转子电阻, 然后根据金属电阻与温度成线性关系的原理, 便可间接计算出 PMSM 定转子绕组的温度。另一方面,

基于参数辨识技术的电机温度在线监测方法无须使用温度传感器等额外设备,可降低硬件设备开销。

针对这一问题,在第4章中研究了一种免疫协同粒子群进化算法,该算法具有大空间快速搜索能力以及动态目标优化能力,适合复杂非线性动态系统建模及参数辨识。基于该算法本章研究了一种基于免疫协同粒子群进化算法的永磁同步电机多参数辨识模型方法。将免疫协同粒子群进化算法应用于永磁同步电机多参数辨识。将该方案在一个永磁同步电机(PMSM)驱动平台上进行了验证,实验结果表明该辨识模型方法不需要依赖任何数据手册上的电机设计值,同时辨识出了电机定子电阻、转子磁链、 d - q 轴电感这几项电机参数;且能跟踪电机各参变量的变化值,算法稳健性较好,避免了复杂数学运算,降低了辨识的复杂性。

7.2 PMSM 数学模型

考虑永磁同步电机具有理想的对称结构,应用中忽略 PMSM 的磁场饱和、铁损和涡流损耗等,其在定子三相坐标系下的状态方程可以表示如下:

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} L_a & M_{ab} & M_{ac} \\ M_{ab} & L_b & M_{bc} \\ M_{ac} & M_{bc} & L_c \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} \quad (7.1)$$

其中

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = \begin{bmatrix} -\omega\psi_m \sin(\theta) \\ -\omega\psi_m \sin(\theta - \frac{2\pi}{3}) \\ -\omega\psi_m \sin(\theta + \frac{2\pi}{3}) \end{bmatrix} \quad (7.2)$$

在式(7.1)中, (u_a, u_b, u_c) , (i_a, i_b, i_c) 分别表示永磁同步电机系统 abc 三相电压和电流; R 、 $(L_a=L_b=L_c=L_s)$ 和 ψ_m 分别表示永磁同步电机系统的定子绕组电阻、自感、转子永磁磁链幅值; ω 和 θ 分别表示电机系统的电机角速度、转子位置角; M_{ab} 、 M_{ac} 、 M_{bc} (其中 $M_{ab}=M_{ac}=M_{bc}=M$) 分别为 ab 相、 ac 相及 bc 相之间的互感。对于 abc 三相定子绕组,通过 Clarke(3/2)变换原理将三相变换成两相,具体如下:

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad (7.3)$$

在式 (7.3) 中, x_a 、 x_b 、 x_c 分别表示 abc 三相定子坐标系中的定子电流/电压。 x_α 、 x_β 分别表示两相静止坐标系中 $\alpha\beta$ 轴的定子电流/电压。电机 abc 三相定子电流/电压信号通过 Clarke 变换到 $\alpha\beta$ 两相静止坐标系中, 如下所示。

$$\begin{bmatrix} \frac{di_\alpha}{dt} \\ \frac{di_\beta}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & 0 \\ 0 & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} + \frac{1}{L} \begin{bmatrix} \psi_m \omega \sin(\theta) + u_\alpha \\ -\psi_m \omega \cos(\theta) + u_\beta \end{bmatrix} \quad (7.4)$$

式 (7.4) 表示隐极式 PMSM 系统 ($L=L_s$) 在 $\alpha\beta$ 两相静止坐标系中的状态方程, (i_α , i_β) 与 (u_α , u_β) 分别表示 $\alpha\beta$ 两相静止坐标系的定子电流和电压。将电机 $\alpha\beta$ 两相静止坐标系电流/电压信号通过 Park 变换原理转换到转子旋转 d - q 轴坐标系中:

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \quad (7.5)$$

在式 (7.5) 中, x_d 、 x_q 分别表示转子旋转坐标系 d - q 轴的定子电流或电压。永磁同步电机在 d - q 轴转子旋转坐标系上的状态方程为

$$\begin{bmatrix} U_d \\ U_q \end{bmatrix} = \begin{bmatrix} R + \frac{di_d}{dt} \cdot L_d & -\omega L_q \\ \omega L_d & R + \frac{di_q}{dt} \cdot L_q \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega \psi \end{bmatrix} \quad (7.6)$$

式 (7.6) 中 (i_d , i_q), (u_d , u_q) 分别表示 d - q 轴定子电流、电压; ω 为电气角度转速; R 、 L_d 、 L_q 、 ψ_m 分别为电机绕组电阻、 d 轴电感、 q 轴电感和永磁磁链。式 (7.6) 的稳态状态离散方程如下:

$$\begin{bmatrix} U_d(k) \\ U_q(k) \end{bmatrix} = \begin{bmatrix} R & -\omega(k)L_q \\ \omega(k) \cdot L_d(k) & R \end{bmatrix} \begin{bmatrix} i_d(k) \\ i_q(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \omega(k) \cdot \psi(k) \end{bmatrix} \quad (7.7)$$

(k) 表示第 k 次采样, 永磁同步电机一般当 $i_d=0$ 时进行解耦控制, $i_d=0$ 的好处在于让所有的电流都用来产生转矩, 其稳态条件下可将式 (7.7) 简

化写成

$$\begin{bmatrix} U_d 0(k_0) \\ U_q 0(k_0) \end{bmatrix} = \begin{bmatrix} R & -\omega(k_0)L_q & 0 & 0 \\ \omega(k_0) \cdot L_d(k_0) & R & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{d0}(k_0) \\ i_{q0}(k_0) \end{bmatrix} + \begin{bmatrix} 0 \\ \psi\omega(k_0) \end{bmatrix} \quad (7.8)$$

$i_d=0$ 实现了 d - q 轴电流解耦, 使定子电流只有 q 轴交流分量, 当 $i_d \neq 0$ 时, 会削弱或者增强磁场的变化, 对于参数辨识来说, $i_d \neq 0$ 时其所辨识参数值会与 $i_d=0$ 时的值有微弱变化。参数集合 $\{R, L_d, L_q, \psi_m\}$ 需要同时辨识, 电机方程的阶数是二, 而待辨识的参数有四个。相对于需要辨识的参数个数来说, 电机方程的阶数是欠缺状态。由于电机的定子电阻, 转子磁链以及绕组电感这些量会随着负载和温度等因素的变化而变化, 使得待辨识量的辨识结果不准确。参考文献[125]通过在短时间内注入一个 $i_d \neq 0$ 的 d 轴电流, 得到一种四阶电机 d - q 轴满秩模型, 如下所示。

$$\begin{bmatrix} U_{d0}(k_0) \\ U_{q0}(k_0) \\ U_d(k_1) \\ U_q(k_1) \end{bmatrix} = \begin{bmatrix} R & -\omega(k_0)L_q & 0 & 0 \\ \omega(k_0) \cdot L_d(k_0) & R & 0 & 0 \\ 0 & 0 & R & -\omega(k_1)L_q \\ 0 & 0 & \omega(k_1) \cdot L_d(k_1) & R \end{bmatrix} \begin{bmatrix} i_{d0}(k_0) \\ i_{q0}(k_0) \\ i_{d1}(k_1) \\ i_{q1}(k_1) \end{bmatrix} + \begin{bmatrix} 0 \\ \psi\omega(k_0) \\ 0 \\ \psi\omega(k_1) \end{bmatrix} \quad (7.9)$$

7.3 基于免疫智能算法的永磁同步电机多参数辨识与温度监测模型

7.3.1 基于 ICPSO 的永磁同步电机多参数辨识思想

对于一个确定模型参数未知系统, 其系统参数辨识问题可归结为一个系统优化问题。辨识的基本原理是通过理论模型输出与实际系统输出的差来决定。针对永磁同步电机系统, 首先研究满秩参考模型设计方案: 对于允许注入扰动信号的应用场合, 可以考虑注入信号来获得满秩参考模型。永磁同步电机系统的多参数辨识问题通过选取数学模型中未知参数的合适值, 使得测量信号与理想模型之间的误差函数值取得最优目标。利用免疫协同进化粒子群进化算法的空间快速并行搜索能力可在问题解空间中搜索到最优解。为了便于免疫协同进化粒子群进化算法进行参数估计, 待辨识系统描述为一个如下微分方程:

$$\begin{aligned} x &= f(p, x(t), u(t)) \\ y(t) &= g(p, x) \end{aligned} \quad (7.10)$$

式中, $x(t)$ 是系统状态变量, $u(t)$ 是系统输入矢量, p 为参数矢量, 即需要辨识的系统参数, 在本书中使用实数编码, $y(t)$ 为系统输出。为了辨识未知参数矢量 p , 同时建立一个期望系统状态模型:

$$\begin{aligned} \hat{x} &= f(\hat{p}, x(\hat{t}), u(t)) \\ y(\hat{t}) &= g(\hat{p}, \hat{x}) \end{aligned} \quad (7.11)$$

由图 7.1 可知基于免疫协同粒子群进化的永磁同步电机系统多参数辨识是在允许参数集中搜索最优的参数矢量集, 使误差适应度函数值达到极小值。适应度函数如下所示:

$$f(\hat{p}) = \sum_{k=1}^n |y(t) - \hat{y}(t)| \quad (7.12)$$

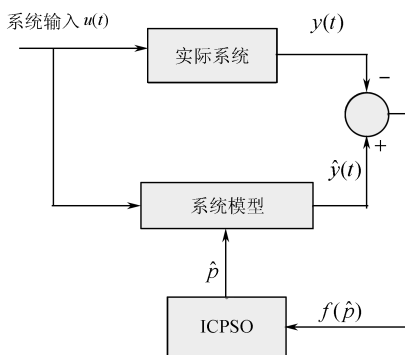


图 7.1 基于 ICPSO 的永磁同步电机系统多参数辨识思想

对于永磁同步电机系统多参数辨识问题, 辨识算法应具有较强的全局和局部动态寻优能力, 免疫协同粒子群进化的永磁同步电机系统多参数辨识思想如下。

步骤 1: 系统相关参数初始化, 并确定系统每个参数矢量的区间约束条件。

- 步骤 2: 根据参数进行系统模型计算, 输出系统矢量, 计算得到基于系统误差的适应度函数值。
- 步骤 3: 免疫协同粒子群进化算法协同操作、进化。
- 步骤 4: 判断算法终止条件是否退出, 否则返回步骤 2 重新计算。

7.3.2 基于 ICPSO 的永磁同步电机多参数辨识与温度监测模型

如图 7.2 所示为基于免疫协同粒子群进化的永磁同步电机多参数辨识与温度监测模型, U_d 和 U_q 被当做永磁同步电机实际模型和理想模型的输入, 免疫协同粒子群进化算法对待辨识参数值进行搜索寻优, 通过实际模型和理想模型输出电压的比较而得到适应度函数, 选择最优适应度值所对应的参数向量, 即所辨识的实际电机参数值, 最后依据定子电阻的变化来实现温度监测。

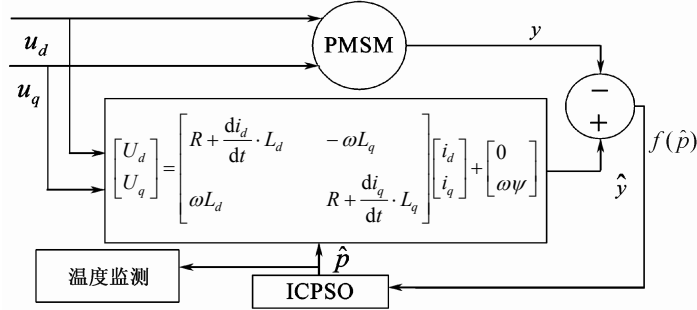


图 7.2 基于免疫协同粒子群进化的永磁同步电机多参数辨识模型

当采样的电流电压值被发送上位 PC 后, 将被传递给基于 Visual C++ 6.0 的 ICPSO 程序以进行 PMSM 多参数辨识, 图 7.3 为 ICPSO 算法的 PMSM 参数辨识与温度监测流程, 其控制原理图如图 7.4 所示。

参数辨识与温度监测算法伪代码如下。

ICPSO-PMSM 参数辨识与温度监测算法 ()

- 步骤 1: 算法相关参数初始化, 电气参数信号输入。
- 步骤 2: 基于 ICPSO 算法的 PMSM 参数估计, 计算可辨识系统状态变量, 输出参数向量, 计算得到误差目标函数。

步骤 3: 目标函数优化计算。

步骤 4: 最优结果选择。

步骤 5: 计算定子绕组温度, 实现温度监测。

步骤 6: 是否达到退出条件, 如果没有执行步骤 2。

步骤 7: 最优结果输出。

步骤 8: 算法结束。

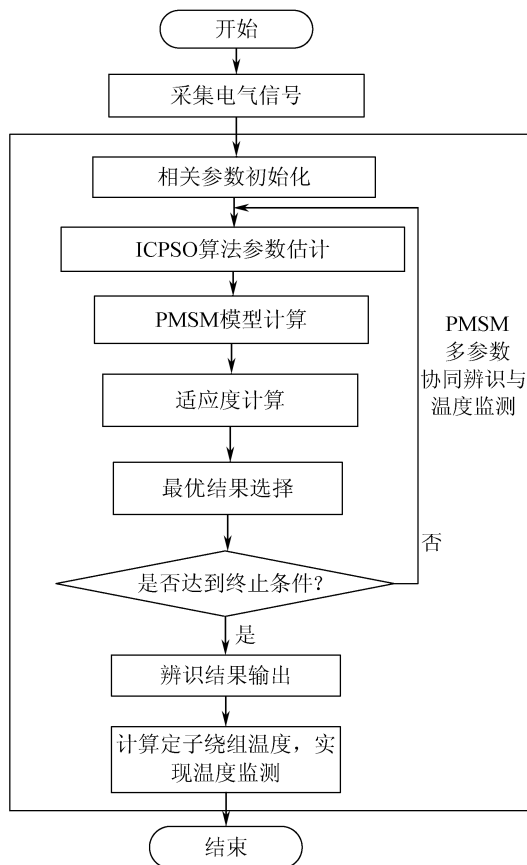


图 7.3 基于 ICPSO 算法的 PMSM 参数辨识与温度监测流程

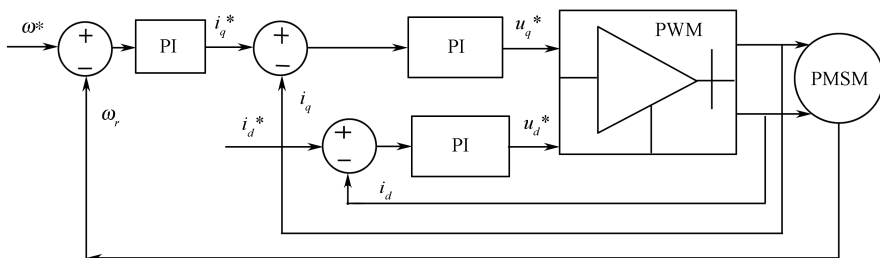


图 7.4 永磁同步电机矢量控制原理图

适应度函数设计如下：

$$\begin{cases} \min F = \frac{\sum_{i=1}^M f(p)}{M} \\ \text{s.t.} \\ R_{\min} \leq R \leq R_{\max} \\ L_{d \min} \leq L_d \leq L_{d \max} \\ L_{q \min} \leq L_q \leq L_{q \max} \\ \psi_{\min} \leq \psi \leq \psi_{\max} \end{cases} \quad (7.13)$$

$$\begin{aligned} f(p) = & \sum_{k=1}^n w_1 |u_{d0}(k) - \hat{u}_{d0}(k)| + w_2 |u_{q0}(k) - \hat{u}_{q0}(k)| + \\ & w_3 |u_d(k) - \hat{u}_d(k)| + w_4 |u_q(k) - \hat{u}_q(k)| \end{aligned} \quad (7.14)$$

式 (7.14) 中 $0 < w_1, w_2, w_3, w_4 < 1$ 为上式 4 个目标函数的权重系数，且 $w_1 + w_2 + w_3 + w_4 = 1$ ，根据文献[195]第 i 个目标函数的权重系数为 $W_i = \frac{f_i}{\sum_{i=1}^n f_i}$ ， f_i

为第 i 个目标函数。在这里可知每个目标都同等重要，所以取 $w_1 = w_2 = w_3 = w_4$ ，系统的参数辨识等同于多维动态函数优化问题，由于实际系统不稳定性及目标函数存在有局部极值点等，常规优化方法难以得到高精度参数估计值，需要寻求一种多样性好、收敛速度快的自适应智能方法来求解此目标函数，常规优化方法难以得到高精度参数估计值，书中采用前些年研究的免疫协同粒子群进化算法求解此动态优化问题。

7.4 实验及分析

7.4.1 实验方案与平台

永磁同步电机参数如表 7.1 所示。

表 7.1 永磁同步电机参数

额定电流	4A
额定转速	400rpm
直流母线电压	36V
测量的定子绕组电阻 ($T=25^{\circ}\text{C}$)	0.330Ω
外部接线电阻	0.043Ω
自感	2.91mH
互感	-0.330mH
d 轴电感	3.24mH
q 轴电感	3.24mH
磁链	77.6mWb
极对数	5

为了验证文中所提辨识模型方法的有效性,图 7.5 是被用来测试的原型电机驱动平台及永磁同步电机驱动系统辨识原理方案。辨识算法所需电流、电压值提取方式如图 7.5 所示,霍尔传感器检测电机稳态工作点时的电流并发送到 DSP 系统中,通过 DSP 内置 AD 转换器对电流进行采样并保存到外部 RAM 存储器中;给定电压值由矢量控制 d - q 轴 PI 调节器输出值与直流母线电压值计算得到。当采样的电流、电压值被传送到上位 PC 后,将采用基于 VC++6.0 平台的 ICPSO 程序对电机绕组电阻、 d - q 轴电感及其磁链进行辨识。辨识后的数据再通过串口传递给电机系统,便于控制。主要利用 PC 的高性能计算能力,PC 的计算能力强于普通嵌入式微型计算机系统,这样基本能实现实时准确辨识。PC 配置为: AMD Athlon(tm) II X2 250 处理器,主频 3.0GHz, 3.25GB 内存, Windows XP, Visual C++6.0。图 7.6~图 7.8 是电机采样电流及其电压转速在 abc 三相定子坐标参考系和 d - q 转子旋转参考坐标系中对应的波形。

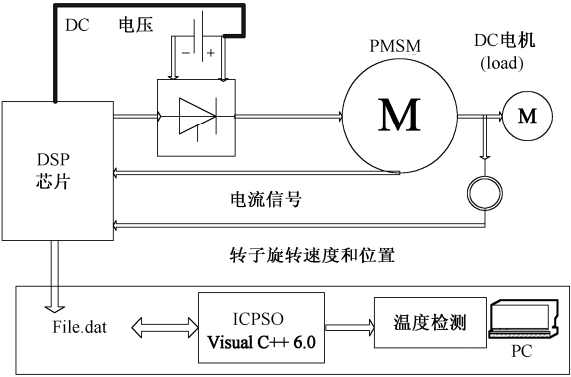


图 7.5 基于 ICPSO 算法的辨识系统

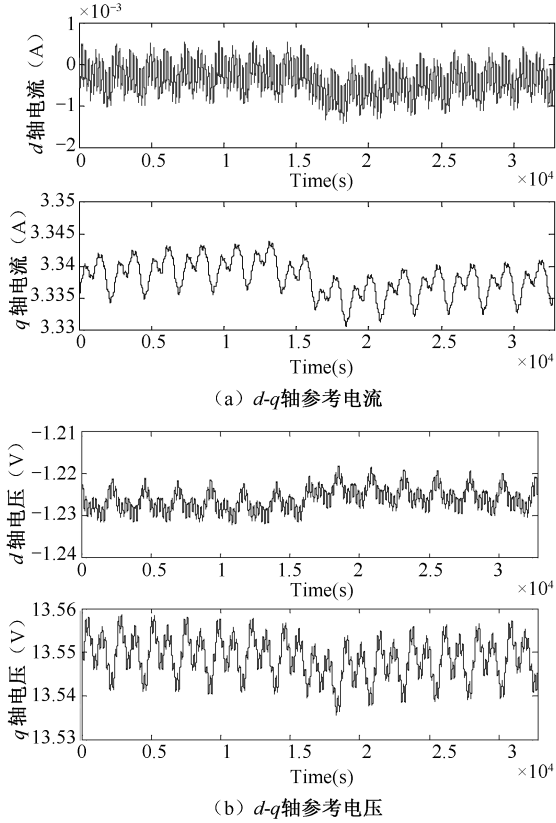


图 7.6 采样电流和电压($i_d=0, i_q=3.5$)

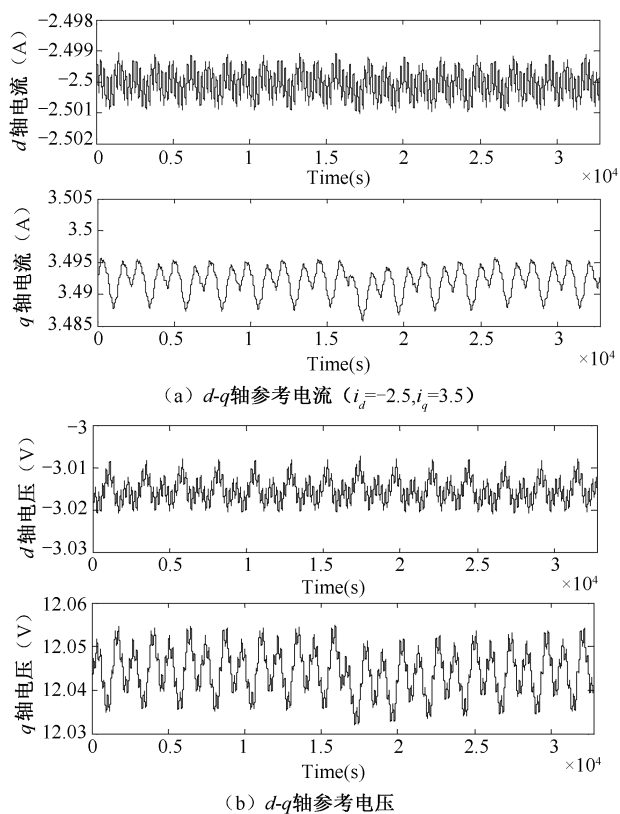


图 7.7 采样电流和电压 ($i_d=-2.5, i_q=3.5$)

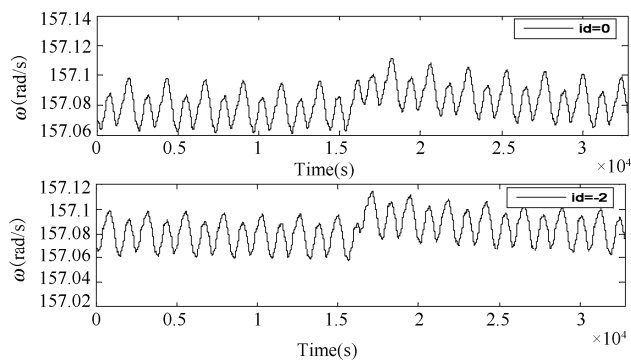


图 7.8 电机转速

7.4.2 实验结果

1. 参数辨识实验结果

由于 PMSM 多参数辨识需要实时性, 最大迭代代数 $T=300$, 所有算法运行 30 次。为了充分证明文中所提方法模型的有效性, 将 ICPSO 算法与多种变种粒子群算法进行比较, 同时也将多种变种粒子群算法应用于永磁同步电机系统多参数辨识。HPSOM 算法表示基于随机均匀变异的粒子群算法^[156], HGAPSO 为基于遗传算法的混合粒子群算法^[141], HPSOWM 为基于小波变异的粒子群算法^[143], CLPSO 为全面学习粒子群算法^[147], A-CLPSO 为自适应全面学习粒子群算法^[148], APSO 表示一种自适应模糊分类粒子群算法^[146]。实验参数设置: HPSOM, HGAPSO, HPSOWM 参数设置参照文献^[143]。CLPSO 算法中 $\omega \in [0.9, 0.4]$, $c_1=c_2=1.49445$; APSO 算法中 $\omega \in [0.9, 0.4]$, $c_1=c_2=2.0$ 。为了保证测试公平, 所有粒子群算法单个种群粒子规模为 50, 文中 ICPSO 算法 $\omega \in [0.91, 0.4]$, $c_1=c_2=1.49445$, 其子种群规模设定为 4。在三种条件下设计实验方案对 PMSM 多参数进行辨识。

实验方案一: 常温条件下的 PMSM 多参数辨识结果。

表 7.2 中 $\{R, L_d, L_q, \psi_m\}$ 分别表示定子绕组电阻、磁链、 d 轴电感、 q 轴电感。从表 7.2 中可以看出文中 ICPSO 算法平均值与标准均方差均优于其他几种变种 PSO 算法。从计算时间来看 ICPSO 在达到稳定时 (平均迭代 200 代数) 为 12.654s, 仅次于 APSO 算法, 优于其他 5 种算法 (HPSOM, HGAPSO, HPSOWM, CLPSO, A-CLPSO)。说明算法 ICPSO 在常温条件下 PMSM 参数辨识具有一定计算机复杂度优势。从 t-value 值可以看出, ICPSO 与其他几种变种粒子群算法的 t-value 值均高于 2.06, ICPSO 算法在常温条件下对 PMSM 多参数辨识效果明显优于其他几种变种 PSO 算法, 与其他算法相比具有 98% 的置信度。常温条件下多种算法的 PMSM 多参数辨识结果适应度收敛曲线如图 7.9 所示。从图 7.10 可以看出本书 ICPSO 辨识曲线值能收敛到一稳态值。其他变种粒子群算法在辨识过程中, PMSM 参数值出现了较大的波动与误收敛现象, 这主要是因为实际电机运行状态非线性程度高, 目标函数存在有局部极值点等, 解搜索空间不大、稳健性不好、全局寻优能力不强的算法难以得到高精度参数估计值。而本书的 ICPSO 方法具有协同并行搜索机制, 信息

交互能力强，同时具有多样性保持机制，整个算法搜索空间大，不易陷入局部极值点，同时算法稳健性较好。所以与其他变种粒子群相比，ICPSO 得到了更好的辨识结果。

表 7.2 常温条件下多种算法的 PMSM 多参数辨识结果比较

$T=300$		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	ICPSO
待辨识参数		辨识结果						
$R(\Omega)$		0.338	0.332	0.363	0.367	0.405	0.372	0.3734
$\psi_m(\text{Wb})$		0.0790	0.0792	0.0783	0.0783	0.0770	0.0783	0.0778
$L_d(\text{H})$		0.00344	0.00275	0.00376	0.00382	0.00380	0.00336	0.0033
$L_q(\text{H})$		0.00395	0.00398	0.00387	0.00388	0.00371	0.00398	0.00396
Time(s)		26.540	16.990	26.181	13.093	13.159	10.634	12.654
Fitness	mean	0.247	1.888	0.844	2.375	2.726	4.163	0.149
	Std.dev	0.182	0.943	0.457	0.890	0.912	2.276	0.028
	t-value	3.7632	13.0341	10.7335	17.6769	19.9710	12.4697	N/A

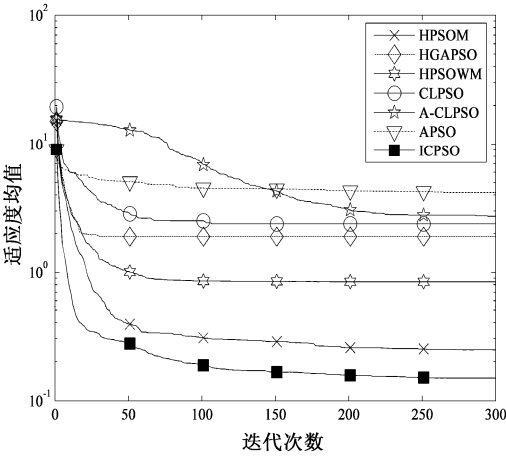
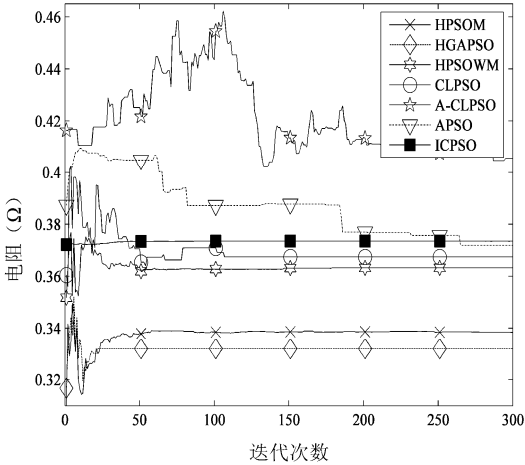


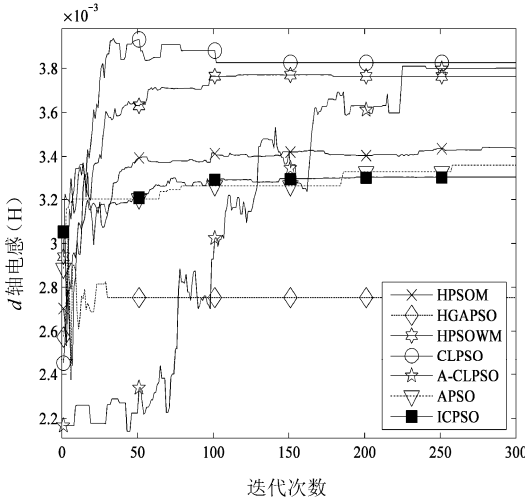
图 7.9 常温条件下多种算法的 PMSM 多参数辨识适应度收敛曲线

从图 7.10 可以看出 ICPSO 算法常温条件下对 PMSM 多参数辨识结果很准确，能有效地辨识电机电阻、 d - q 轴电感与转子磁链，且收敛速度较快，所以估算出来的电阻值 (0.3734Ω) 几乎等于用 LCR 电桥测得的实际值

(0.373Ω)。转子磁链值 (77.8mWb) 和设计手册里的转子磁链值 (77.6mWb) 大小几乎一致。但与电机手册设计值略有一些微小差别，其主要原因在于电机运行过程中，存在各种系统扰动，其工况有所改变。

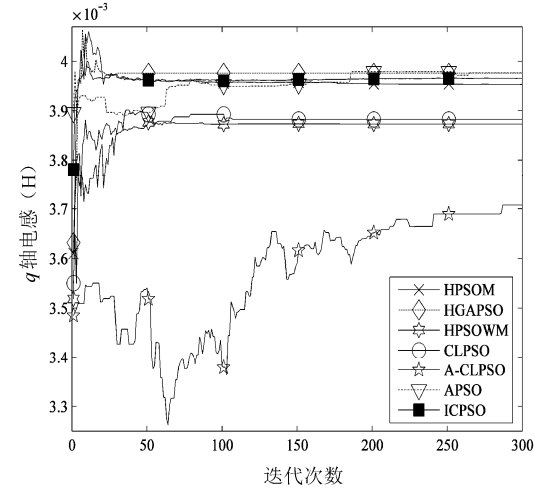


(a) 电机绕组电阻辨识曲线

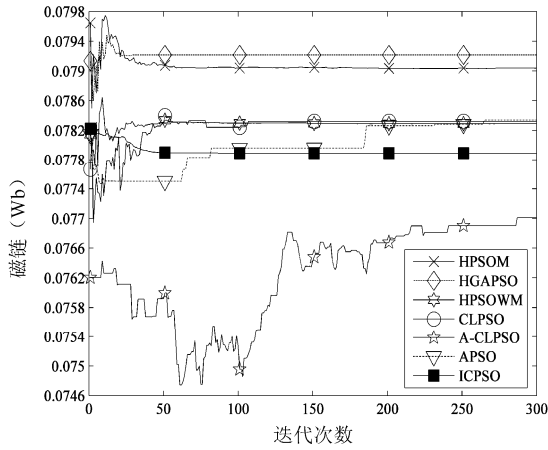


(b) d 轴电感辨识曲线

图 7.10 常温条件下 PMSM 多参数辨识值



(c) q 轴电感辨识曲线



(d) 永磁磁链辨识曲线

图 7.10 常温条件下 PMSM 多参数辨识值 (续)

实验方案二：加热条件下的 PMSM 多参数辨识结果 (见表 7.3)。

表 7.3 加热下条件下多种算法的 PMSM 多参数辨识结果比较

$T=300$		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	ICPSO
待辨识参数		辨识结果						
$R(\Omega)$		0.453	0.459	0.458	0.473	0.479	0.455	0.453
$\psi_m(\text{Wb})$		0.0769	0.0770	0.0768	0.0764	0.0762	0.0772	0.0770
$L_d(\text{H})$		0.00347	0.00341	0.00411	0.00414	0.00388	0.00370	0.00364
$L_q(\text{H})$		0.00376	0.00378	0.00376	0.00370	0.00369	0.00386	0.00376
Time(s)		26.569	17.006	26.217	13.172	13.128	11.313	13.445
Fitness	mean	0.283	2.246	0.410	1.968	1.135	3.300	0.261
	Std.dev	0.121	1.121	0.215	0.841	0.483	3.866	0.147
	t-value	0.8171	12.4147	4.0453	14.1380	12.2409	5.5544	0.8171

为了验证电机在温度变化条件下, ICPSO 算法能否有效跟踪永磁同步电机参数变化, 用热风枪持续加热电机 20 分钟后对永磁同步电机多参数进行辨识。

表 7.3 为加热条件下多种算法的 PMSM 多参数辨识结果, 从表 7.3 中可以看出文中 ICPSO 算法适应度平均值优于其他几种变种 PSO 算法。从计算平均时间来看 ICPSO 在达到稳定时(平均迭代 210 代数)为 13.445s, 仅次于 APSO、CLPSO、A-CLPSO 算法, 优于其他三种算法(HPSOM, HGAPSO, HPSOWM)的计算时间复杂度。说明算法 ICPSO 在加热条件 PMSM 参数辨识具有一定计算复杂度优势。同时, 从 t-value 值可以看出, ICPSO 与 HPSOM 的 t-value 等于 0.8171, 表明 ICPSO 优于 HPSOM 具有 75%的置信度。其他 5 种变种粒子群算法的 t-value 值均高于 2.06, ICPSO 算法在常温条件下对 PMSM 多参数辨识效果明显优于其他 5 种变种 PSO 算法, 具有 98%的置信度。从图 7.11 和图 7.12 可以看出 ICPSO 算法所辨识的参数较为稳定, 同时具有很好的动态跟踪性能。电阻值由于温升从 0.373Ω 上升到 0.453Ω , 同时转子永磁磁链幅值由于温升作用由 77.8mWb 降到到 77.0mWb , d - q 轴电感也将相应改变, 但变化不大, 温度升高电阻值必然升高, 由式 (7.8) 可知电阻增大将引起磁场密度降低, 这一趋势是正确的。由辨识结果可知, 本书方法 ICPSO 算法能够很好跟踪电机定转子温度的变化。

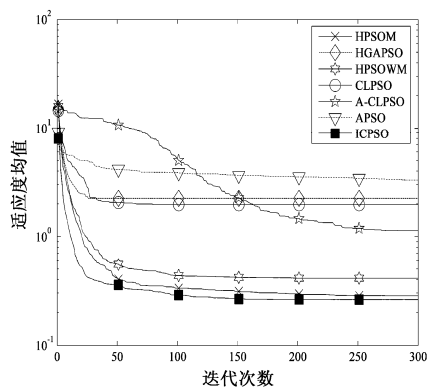
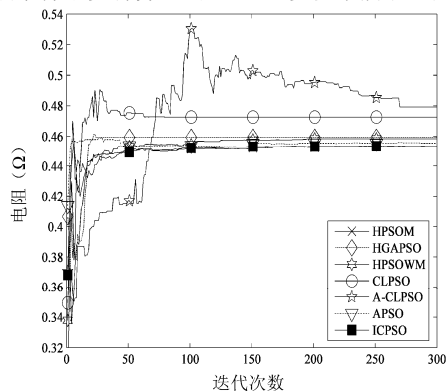
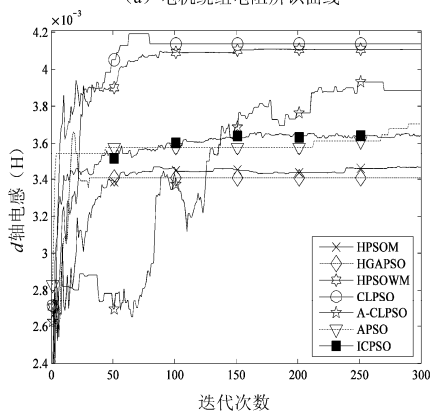


图 7.11 加热条件下多种算法的 PMSM 多参数辨识适应度收敛曲线

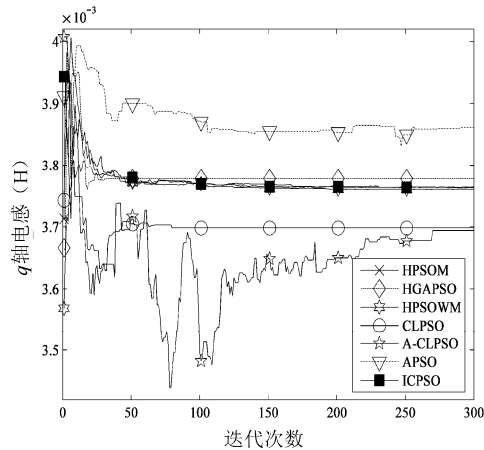


(a) 电机绕组电阻辨识曲线

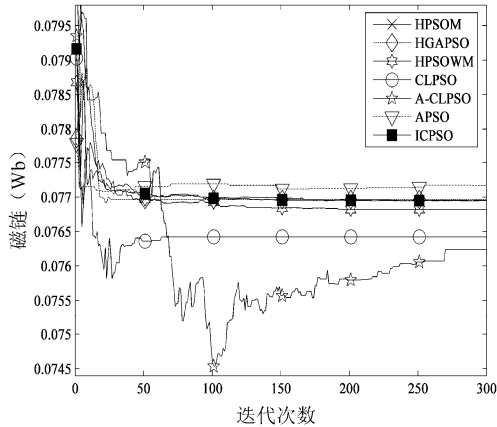


(b) d 轴电感辨识曲线

图 7.12 加热条件下 PMSM 多参数辨识值



(c) q 轴电感辨识曲线



(d) 永磁磁链辨识曲线

图 7.12 加热条件下 PMSM 多参数辨识值 (续)

从图 7.12 可以看出在加热条件下本书 ICPSO 辨识值能收敛到稳定值, 其他变种粒子群算法在辨识过程中, 所辨识 PMSM 参数值出现了较大的波动及误收敛现象, 这主要是因为实际电机运行状态随着温度的变化而呈现强非线性, 目标函数存在有局部极值点等, 算法寻优能力不强难以得到高精度参数估计值。而本书的 ICPSO 方法具有协同并行搜索机制, 信息交互能力强, 具有多样性保持机制, 整个算法搜索空间大, 不易陷入局部极值点, 同时算

法稳健性较好，具有较好的动态跟踪能力，能跟踪 PMSM 变化的参数值。

实验方案三：外串电阻条件下的 PMSM 多参数辨识结果。

为了验证外加电阻干扰的情况下算法能否辨识电机系统参数变化，特别是对电阻的准确跟踪。在定子三相绕组串入电阻 ($R_p=0.414\Omega$) 后的辨识结果如表 7.4、图 7.13、图 7.14 所示， R_p 是串入电阻的阻值。

表 7.4 外串电阻条件下多种算法的 PMSM 多参数辨识结果比较

$T=300$		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	ICPSO
待辨识参数		辨识结果						
$R+R_p(\Omega)$	$R_p=0.414\Omega$	0.801	0.770	0.799	0.800	0.829	0.731	0.785
$\psi_m(\text{Wb})$		0.0773	0.0781	0.774	0.0774	0.0766	0.0794	0.0777
$L_d(\text{H})$		0.00387	0.00327	0.00389	0.00380	0.00355	0.00363	0.00362
$L_q(\text{H})$		0.00359	0.00366	0.00361	0.00366	0.00355	0.00385	0.00365
Fitness	Mean	0.459	2.486	0.463	3.620	1.824	4.464	0.194
	Std.dev	0.335	1.282	0.348	1.667	0.785	2.992	0.073
	t-value	5.4653	12.6214	5.3494	14.5185	14.6195	10.0884	0

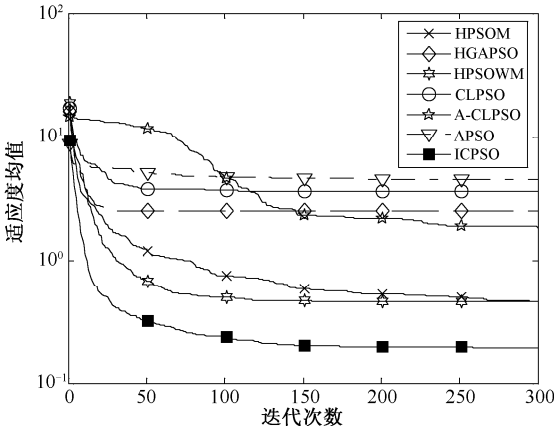


图 7.13 外串电阻条件下多种算法的 PMSM 多参数辨识适应度收敛曲线

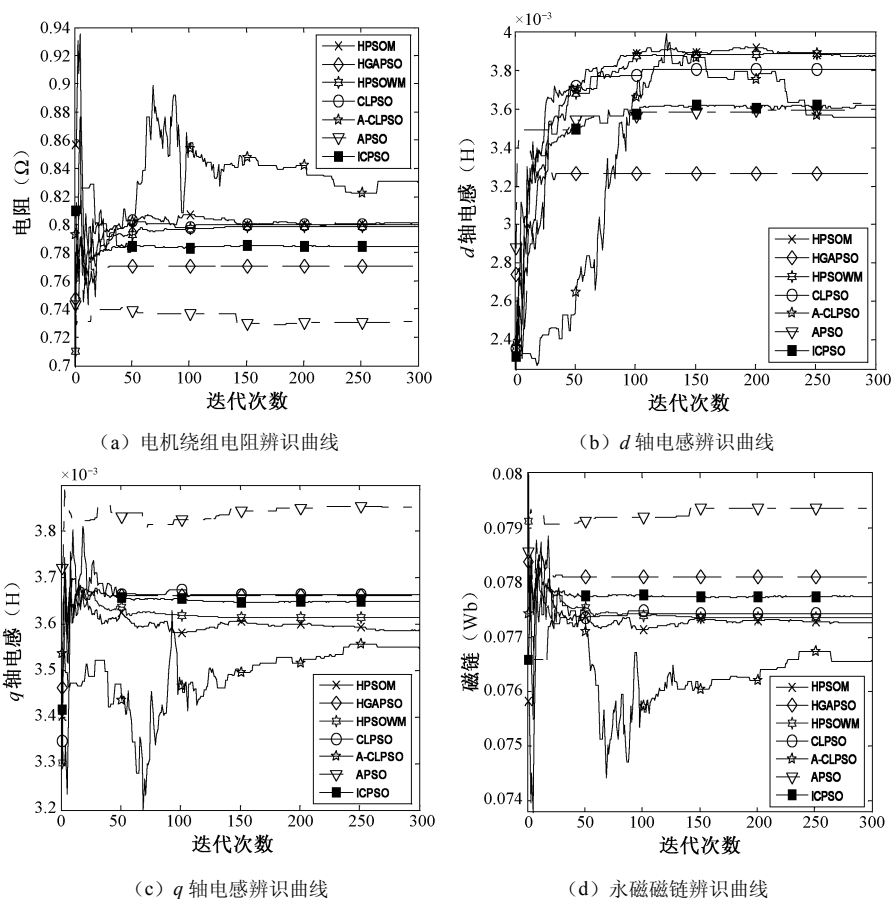


图 7.14 外串电阻条件下 PMSM 多参数辨识值

从表 7.4 中可以看出文中 ICPSO 算法适应度平均值优于其他几种变种 PSO 算法, 观察辨识得到的电阻变化曲线, 表 7.5 中辨识得到的电阻变化量 ($0.785-0.373=0.412\Omega$) 和串入电阻大小 (0.414Ω) 几乎一致, 证明了 ICPSO 能准确跟踪电阻的变化。

2. 电机定子绕组温度在线监测

金属除了在绝对零度及熔点附近外, 其电阻与其温度之间存在严格的线性关系, 其关系数学表达式如下式所示:

$$R = R_0[1 + \alpha(T - T_0)] \quad (7.15)$$

表 7.5 基于 ICPSO 的 PMSM 多参数结果对比

参数	常温	加热 20 分钟	串入电阻
$R(\Omega)$	0.3734	0.453	0.785
$\Psi_m(\text{Wb})$	0.0778	0.0770	0.0777
$L_d(\text{H})$	0.0033	0.00364	0.00362
$L_q(\text{H})$	0.00396	0.00376	0.00365

式 (7.15) 中, R 为温度 T 下的金属电阻, R_0 为参考温度 T_0 下的金属电阻, a 为电阻温度系数, 对于金属铜, a 为 0.004。在算法辨识出 PMSM 转子电阻之后, 根据上式便可计算出定转子温度, 从而实现永磁同步电机定转子温度在线监测, 基于 ICPSO 算法的 PMSM 的温度监测曲线如图 7.15 所示。图 7.15 (a) 为持续加热 20 分钟后的温度曲线, 图 7.15 (b) 为加热 20 分钟后冷却 9 分钟的温度曲线。从图 7.15 (a) 可以得出加热初始时由于受热不均电阻快速升高, 所以温度较高, 后期电阻值趋于稳定, 所以后期温度趋于平稳。从图 7.15 (b) 可以看出电阻加热后冷却过程中, 温度曲线呈非线性下降趋势, 这一现象满足金属物理退温过程为非线性递减退火过程。以上实验结果可以得出基于 ICPSO 参数辨识技术的永磁同步电机定转子温度在线监测方法是可行的。

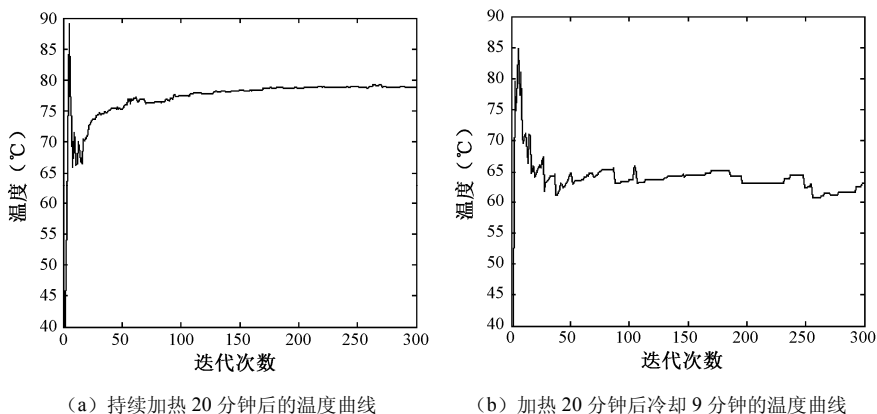


图 7.15 基于 ICPSO 算法的 PMSM 定子绕组温度监测曲线

7.5 本章小结

本章首先简述了永磁同步电机系统参数辨识现状,文中研究的免疫协同粒子群进化算法具有大解空间快速并行搜索能力及动态问题求解性能,适合于复杂非线性动态系统建模与参数辨识。将免疫协同粒子群进化算法应用于永磁同步电机系统多参数辨识与温度监测中,提出了一种基于免疫协同粒子群进化算法的永磁同步电机系统多参数辨识与状态监测模型方法,同时将多种改进的粒子群算法应用到永磁同步电机参数辨识与温度监测中。最后将免疫协同粒子群进化算法应用于永磁同步电机参数辨识与温度监测;在永磁同步电机平台上实验结果表明该方法不需要知道电机设计参数先验知识,能够有效地辨识电机电阻、 $d-q$ 轴电感与转子磁链。依据定子绕组随温度变化而呈线性关系,即可间接计算定转子温度,从而实现永磁同步电机系统温度在线监测。从实验结果与分析来看,本文所提方法适合永磁同步电机多参数辨识与温度监测。同时本文提出的永磁同步电机多参数辨识技术可应用于电动机状态监测与故障诊断系统,可降低额外的硬件设备开销。本理论方法成果将为这类永磁同步电驱动系统参数辨识的多参数辨识和状态估计技术提供一个新的求解思路,并将大大推动该类系统相关技术的发展。

后 记

现代智能算法理论及应用是国际上十分活跃的研究课题。人工免疫系统模拟生物免疫系统进化行为的智能特征，具有自组织、自学习能力，具有解决复杂优化问题的优点。现代工业系统变得越来越复杂，而复杂系统的建模、优化与控制需要高性能的算法来辅助，依靠单一模式的求解方法难以满足系统性能要求。混合免疫智能处理技术为这类问题提供了有效的途径，同时也是人工免疫系统研究的发展方向。

基于免疫系统的机理，深入挖掘生物免疫系统中蕴含的智能学习机制并结合其他智能处理方法的优点，全书研究了几类混合免疫智能优化算法及其相关应用，从算法理论、算法设计、性能测试、比较分析到实际应用展开一系列工作。研究工作的主要结论可归结如下。

① 引入生态学中的协同进化 Lotka-Volterra 思想到人工免疫算法中，考虑了群体间的竞争合作关系，构造了一种竞争合作型协同进化免疫克隆选择模型。各子种群内部通过局部最优免疫优势、克隆扩增和动态高频变异等相关算子操作，运用信息熵理论改善种群多样性，所有子种群共享经过免疫杂交提升操作的高层记忆库，通过迁移操作实现整个种群信息共享与协同进化。

② 为了扩大解的搜索空间，将粒子群体分为捕食与探索两种模态，建立一种免疫双态粒子群优化方法。对处于捕食状态的精英粒子采用精英学习策略，对处于探索状态的微粒采用探索策略，对微粒个体极值进行免疫克隆优化，对不活跃个体进行免疫受体编辑。算法兼顾了抑制早熟停滞现象和避免冗余迭代。

③ 融合免疫系统优化原理、协同进化思想及粒子群的邻域信息,构建免疫协同粒子群进化模型。算法采用并行计算框架,整个群体由记忆种群与若干个普通种群构成。普通微粒种群内部通过精英粒子保留、免疫网络及柯西变异等混合策略共同演化新个体,微粒个体极值采用自适应小波学习以加快收敛速度,免疫克隆选择算法对记忆库进行精细搜索,信息交互机制促进信息共享,有效降低了算法的冗余迭代,扩大了算法解空间搜索范围,提高了对复杂问题的优化能力。

④ 利用克隆选择算法与蚁群算法各自的优势,研究了两类免疫克隆选择与蚁群自适应融合优化模型。第一类,研究了一种基于统计学习的免疫蚁群自适应融合算法,引入混沌扰动增加抗体种群的多样性,通过克隆扩增、免疫基因等相关算子的操作增强了克隆选择算法搜索的效率;自适应统计学习控制参数实现了克隆选择与蚁群优化的有机结合及局部最优搜索策略的应用,克服了抗体种群“早熟”问题,提高了求解精度。第二类,模拟真实蚁群社会中的所有蚂蚁各司其职,又相互依赖协作,形成一个有机整体的客观现象,并结合免疫机理,构造了双态免疫优势蚁群算法模型。该算法将蚂蚁分成两种状态,不同种类的蚁群有不同的信息素调控机制,这样更能够模拟蚁群社会中的蚂蚁有组织,有分工的特点,这种分工组织方式对蚁群完成复杂任务具有十分重要的作用。这样扩大了解的搜索空间,有效抑制了收敛过程中的早熟停滞现象。

⑤ 应用免疫双态粒子群算法对自抗扰控制器进行优化设计。其一,将免疫双态粒子群算法应用于混沌系统自抗扰优化控制中,对自抗扰控制器参数进行优化,应用于混沌系统控制,构建一种基于免疫双态粒子群算法的混沌系统自抗扰优化控制器;其二,利用自抗扰控制器(ADRC)与小脑神经网络(CMAC)各自的优势构造 ADRC-CMAC 并行控制器,利用免疫双态粒子群算法对 ADRC-CMAC 控制器参数进行自学习寻优,构造出一种自抗扰神经网络并行优化控制方法。针对离散混沌系统研究结果表明,以上两种控制方法具有更好的控制性能和较强的鲁棒性。

⑥ 构造了一种基于免疫协同粒子群进化算法的永磁同步电机多参数辨识模型方法。永磁同步电机参数辨识结果表明该方法不需要知道电机设计参数先验知识,能够有效地辨识电机电阻、d-q 轴电感与转子磁链。同时,当电机参数发生变化时,该方法依然能够有效地追踪该参数变化值。在辨识出

电机定子绕组阻值后,根据金属阻值与温度之间的线性原理间接计算定转子温度,从而实现永磁同步电机系统温度在线监测。

全书研究工作取得了一些创新性的成果,为混合免疫智能算法理论与应用的研究提供了一些新的思路,但本身也存在诸多不足,混合免疫智能算法理论及其应用研究也有待更多的研究人员展开更为深入的研究和探索。基于现有的研究基础以及免疫计算智能的发展,未来有望进一步开展的研究工作如下。

1. 建立混合免疫智能算法的统一框架理论体系

一直以来针对人工免疫系统、混合免疫进化算法的研究还处于一种比较分散的研究状态,混合免疫智能算法的认识方面,以及在算法具体实现方面都还没有形成比较统一的框架理论体系。如何从生态进化原理的共同性、特殊性,以及计算机实现算子的相似性与特殊性中提炼出关键元素来建立混合免疫进化算法的统一框架。如何在统一框架下研究混合免疫算法进化策略,并对算法的收敛性、收敛速度、计算复杂性、稳健性等进行理论分析,进一步推动混合免疫智能优化理论与算法的发展。

2. 面向多核/GPU 异构并行系统的并行混合免疫智能算法研究

混合免疫智能算法天生具有的伪并行性特点使其非常适合于算法的并行实现。而随着多个处理器并行计算机的发展及 GPU 通用计算的推广为并行混合免疫智能算法的并行化提供了技术支撑,利用并行计算机来研究混合免疫智能算法可以提高算法的收敛速度,降低算法的实际时间复杂度,提高算法的实时响应。混合免疫智能算法的并行化工程是下一步的研究方向。

3. 拓宽混合免疫智能算法的实际应用研究领域

混合免疫智能算法具有比单一模式的优化方法更强的优势,然而,现实工业系统变得越来越复杂,而复杂系统的优化与控制需要高性能的算法来辅助,依靠传统单一求解方法难以满足系统性能要求。混合免疫智能处理技术能为这类复杂问题系统建模及优化控制提供高效的技术支撑。将混合免疫智能算法结合实际工业应用问题展开研究,将是一个很有前途的发展方向。

参考文献

- [1] Min Gui, Anil Pahwa, Sanjoy Das. Analysis of Animal-Related Outages in Overhead Distribution Systems with Wavelet Decomposition and Immune Systems-Based Neural Networks. IEEE Transactions on Power Systems, 2009, 24(4): 1765-1771.
- [2] Muhammad Rahmat Widyanto, Benyamin Kusumoputro, Hajime Nobuhara, et al. A Fuzzy-Similarity-Based Self-Organized Network Inspired by Immune Algorithm for Three-Mixture-Fragrance Recognition. IEEE Transactions on Industrial Electronics, 2006, 53(1): 313-321.
- [3] Cai KaiYuan, Lei Zhang. Fuzzy Reasoning as a Control Problem. IEEE Transactions on Fuzzy Systems, 2008, 16(3): 600-614.
- [4] Licheng Jiao, Lei Wang. A Novel Genetic Algorithm Based on Immunity. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2000, 30(3): 552-561.
- [5] Ashish Ahuja, Sanjoy Das, Anil Pahwa. An AIS-ACO Hybrid Approach for Multi-Objective Distribution System Reconfiguration. IEEE Transactions on Power Systems, 2007, 22(3): 1101-1112.
- [6] Marco Dorigo, Gianni Di Caro. Ant Algorithms for Discrete Optimization. Artificial Life, 1999, 5(3): 137-172.
- [7] Hong-Wei Ge, Liang Sun, Yan-Chun Liang, et al. An Effective PSO and AIS-Based Hybrid Intelligent Algorithm for Job-Shop Scheduling. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and

- Humans, 2008, 38(2): 358-368.
- [8] 焦李成, 杜海峰, 刘芳等. 免疫优化计算-学习与识别. 北京: 科学出版社, 2007.
- [9] Jerne N. K. Towards a network theory of the immune system. Annual Immunology, 1974, 125C: 373-389.
- [10] Dipankar Dasgupta, Advances in artificial immune systems. IEEE Computational Intelligence Magazine, 2006, 1(4): 40-49.
- [11] De Castro L. N, Von Zuben F. J. Learning and optimization using the clonal selection principle. IEEE Trans on Evolutionary Computation, 2002, 6(1): 239-251.
- [12] 戚玉涛, 刘芳, 焦李成. 基于分布式人工免疫算法的数值优化. 电子学报, 2009, 37(7): 1554-1561.
- [13] 薛文涛, 吴晓蓓, 徐志良. 用于多峰函数优化的免疫粒子群网络算法. 系统工程与电子技术, 2009, 31(3): 705-709.
- [14] 戚玉涛, 刘芳, 焦李成. 基于信息素模因的免疫克隆选择函数优化. 计算机研究与发展, 2008, 45(6): 991-997.
- [15] 余航, 焦李成, 公茂果等. 基于正交试验设计的克隆选择函数优化. 软件学报, 2010, 21(5): 950-967.
- [16] 戚玉涛, 焦李成, 刘芳. 基于并行人工免疫算法的大规模 TSP 问题求解. 电子学报, 2008, 36(8): 1552-1558.
- [17] Anna Szwedicka, Franciszek Seredynski, Albert Y. Zomaya. Multiprocessor Scheduling and Rescheduling with Use of Cellular Automata and Artificial Immune System Support. IEEE Transactions on Parallel and Distributed Systems, 2006, 17(3): 253-262.
- [18] Licheng Jiao, Yangyang Li, Maoguo Gong, et al. Quantum-Inspired Immune Clonal Algorithm for Global Optimization. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics, 2008, 38(5): 1234-1433.
- [19] Felipe Campel, Frederico G. Guimarães, Hajime Igarashi. Multiobjective Optimization Using Compromise Programming and an Immune Algorithm. IEEE Transactions on Magnetics, 2008, 44(6): 982-985.

- [20] Aldo Canova, Fabio Freschi, Michele Tartaglia. Multiobjective Optimization of Parallel Cable Layout, IEEE Transactions on Magnetics, 2007, 43(10): 3914-3920.
- [21] Xiong Hao, Sun Cai-xin. Artificial Immune Network Classification Algorithm for Fault Diagnosis of Power Transformer. IEEE Transactions on Power Delivery, 2007, 22(2): 930-935.
- [22] Slavisa Sarafijanovic, Jean-Yves Le Boudec. An Artificial Immune System Approach With Secondary Response for Misbehavior Detection in Mobile ad hoc Networks. IEEE Transactions on Neural Networks, 2005, 16(5): 1076-1087.
- [23] Rogerio de Lemos, Jon Timmis, Modupe Ayara, et al. Immune-Inspired Adaptable Error Detection for Automated Teller Machines. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2007, 37(5): 873-886.
- [24] Forrest S, Perelson A S, Allen L, et al. Self-nonself discrimination in a computer. In: Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy. Oakland, USA: IEEE Press, 1994:202-212.
- [25] Paul K. Harmer, Paul D. Williams, Gregg H. et al. An Artificial Immune System Architecture for Computer Security Applications. IEEE Trans on Evolutionary Computation, 2002, 6(3): 252-290.
- [26] Luther K, Bye R, Alpcan T, et al. A cooperative AIS framework for intrusion detection. In: Proceedings of IEEE International Conference on Communications. Washington D.C., USA: IEEE Press, 2007:1409-1416.
- [27] 徐雪松, 诸静. 基于免疫识别原理和 RFNN 的复杂系统辨识. 模式识别与人工智能, 2003, 12(4): 397-401.
- [28] 葛宏伟, 李小琳, 梁艳春等. 基于免疫粒子群优化的一种动态递归神经网络辨识与控制非线性系统. 吉林大学学报(工学版), 2008, 7(4): 858-864.
- [29] 付华, 乔德浩, 池继辉. 一种非线性系统参数辨识的耦合算法研究. 西安交通大学学报, 2011, 45(2): 49-53.

- [30] 林金星, 沈炯, 李益国. 基于免疫原理的径向基函数网络在线学习算法及其在热工过程大范围工况建模中的应用. 中国电机工程学报, 2006, 5(9): 14-19.
- [31] Cheng-Hung Chen, Cheng-Jian Lin, Chin-Teng Lin. Using an Efficient Immune Symbiotic Evolution Learning for Compensatory Neuro-Fuzzy Controller. IEEE Transactions on fuzzy systems, 2009, 17(3): 668-682.
- [32] Amanda M. Whitbrook, Uwe Aickelin, et al. Idiotypic Immune Networks in Mobile-Robot Control. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics, 2007, 37(6): 1581-1598.
- [33] Henry Y. K. Lau, Vicky W. K. Wong. An Immunity-Based Distributed Multiagent-Control Framework. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2006, 36(1): 91-108.
- [34] Ying-Xin Liao, Jin-Hua She, Min Wu. Integrated Hybrid-PSO and Fuzzy-NN Decoupling Control for Temperature of Reheating Furnace. IEEE Transactions on Industrial Electronics, 2009, 56(7): 2704-2714.
- [35] 江全元, 邹振宇, 曹一家. 基于免疫遗传算法的多机电力系统 UPFC 控制器. 电工技术学报, 2006, 21(7): 60-68.
- [36] G-C. Liao. Application of an immune algorithm to the shortterm unit commitment problem in power system operation. IEE Proc.-Gener. Transm. Distrib, 2006, 153(3): 308-319.
- [37] 李玉龙, 宗伟, 占鲜艳等. 基于抗体浓度调节新定义下的免疫遗传算法在电压无功优化中的应用. 电工技术学报, 2008, 23(2): 115-119.
- [38] Alex A. Freitas, Jon Timmis. Revisiting the Foundations of Artificial Immune Systems for Data Mining. IEEE Trans on Evolutionary Computation, 2007, 11(4): 521-540.
- [39] Tien Dung Do, Siu Cheung Hui, A. C. M. Fong, et al. Associative Classification With Artificial Immune System. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 217-238.
- [40] Seralozs, en, Salih Gunes, Sadk Kara, et al. Use of Kernel Functions in Artificial Immune Systems for the Nonlinear Classification Problems.

- IEEE transactions on information technology in biomedicine, 2009, 13(4): 621-628.
- [41] Vasileios K. Karakasis, Andreas Stafylopatis. Efficient Evolution of Accurate Classification Rules Using a Combination of Gene Expression Programming and Clonal Selection. IEEE Transactions on Evolutionary Computation, 2008, 12(6): 662-678.
- [42] Yanfei Zhong, Liangpei Zhang, Jianya Gong, et al. A Supervised Artificial Immune Classifier for Remote-Sensing Imagery. IEEE Transactions on Geoscience and Remote Sensing, 2007, 45(12): 3957-3966.
- [43] 马文萍, 焦李成, 张向荣等. 基于量子克隆优化的 SAR 图像分类. 电子学报, 2007, 35(12): 2241-2246.
- [44] 郭建民, 刘石, 姜凡. 模糊免疫网络算法在数字图像火焰监测中的应用. 中国电机工程学报, 2007, 27(2): 1-6.
- [45] 王孙安, 郭子龙. 混沌免疫优化组合算法. 控制与决策, 2006, 21(2): 205-211.
- [46] 刘朝华, 张英杰, 吴建辉. 一种求解 TSP 问题的改进克隆选择算法. 系统仿真学报, 2010, 22(7): 1627-1631.
- [47] 林学颜, 张玲. 现代细胞与分子免疫学. 北京: 科学出版社, 1999, 3-50.
- [48] 史铁鬃. 协和内分泌和代谢学. 北京: 科学出版社, 1999, 6-108.
- [49] Gong MG, Du H. F, Jiao L. C. Optimal approximation of linear systems by artificial immune response. Science in China (Series F-Information Sciences), 2006, 49(1): 63-79.
- [50] Jerne N. K. Towards a Network Theory of the Immune System. Annual Immunology, 1974, 125(C): 373-389.
- [51] Perelson A. S. Immune Network Theory. Immunological review, 1986, (10): 5-3.
- [52] Farmer J. D, Packard N. H, Perelson A. S. The Immune System, Adaptation and Machine Learning. Physica, 1986, 22(D): 187-204.
- [53] De Castro L. N, Zuben F. J. V. Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 239-251.

- [54] Kim J,Bentley P.Towards an Artificial Immune System for NetworkIntrusion Detection: An Investigation of Dynamic Clonal Selection,In Proceedings of 2002 Congresson Evolutionarycomputation. Honolulu:IEEE press,2002:1015-1020.
- [55] Wolfam S.Approaches to complexity engineering.Phisyca D,1986, 22:385-399.
- [56] 靳蕃. 神经网络与神经计算机原理应用. 成都: 西南交通大学出版社, 1991: 4-130.
- [57] 莫宏伟. 人工免疫系统原理及其应用. 哈尔滨: 哈尔滨工业大学出版社, 2002: 5-165.
- [58] 肖人彬, 曹鹏林, 刘勇. 工程免疫计算. 北京: 科学出版社, 2007: 5-128.
- [59] 莫宏伟, 左兴权, 毕晓君. 人工免疫系统研究进展. 智能系统学报, 2009, 4(1): 21-29.
- [60] D. H. Wolpert, W. G. Macready. No free lunch theorems for optimization. IEEE Trans. Evol. Comput., 1997,1(1): 67-82.
- [61] 陈曦, 谭冠政, 江斌. 基于免疫遗传算法的移动机器人实时最优路径规划. 中南大学学报(自然科学版), 2008, 39(3): 577-584.
- [62] 杨东勇, 陈晋音. 基于多种群遗传算法的检测器生成算法研究. 自动化学报, 2009, 35(4): 425-432.
- [63] 葛宏伟, 梁艳春. 基于隐马尔可夫模型和免疫粒子群优化的多序列比对算法. 计算机研究与发展, 2006, 43 (8): 1330-1336.
- [64] 陶新民, 徐晶, 杨立标. 改进的多种群协同进化粒子群优化算法. 控制与决策, 2009, 24(9): 1406-1411.
- [65] Lu Hong, Zhi-Cheng Ji, Cheng-Long Gong. Study on Immune PSO Hybrid Optimization Algorithm, Chinese Conference on Pattern Recognition, 2009. CCPR 2009, Nanjing ,china:IEEE press, 2009:1-4.
- [66] 彭春华. 基于免疫 BPSO 算法与拓扑可观性的 PMU 最优配置. 电工技术学报, 2008, 23(6): 119-124.
- [67] 丛琳, 焦李成, 沙宇恒. 正交免疫克隆粒子群多目标优化算法. 电子与信息学报, 2008, 30(10): 2320-2324.
- [68] 刘丽, 须文波, 吴小俊. 基于全局粒子群的协作型人工免疫网络优化

- 算法. 模式识别与人工智能, 2009, 22(4): 653-660.
- [69] 李中华, 张雨浓, 谭洪舟等. 一类具有精英学习能力的增强型人工免疫网络优化算法. 控制理论与应用, 2009, 26(3): 283-290.
- [70] Ling Q, Yi X. C, Jian L.L, et al. Diversity Guaranteed Ant Colony Algorithm Based on Immune Strategy. In Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06). Hanzhou, china: IEEE Press, 2006: 217-223.
- [71] 刘朝华, 李小花, 章兢. 精英免疫克隆选择的协同进化粒子群算法. 电子学报, 2013, 41(11): 2167-2173.
- [72] 夏虎, 庄健, 王立忠等. 一种考虑环境作用的协同免疫遗传算法. 西安交通大学学报, 2009, 43(11): 80-88.
- [73] 王磊, 刘小勇. 协同人工免疫计算模型的研究. 电子学报, 2009, 37(8): 1739-1745.
- [74] 吴秋逸, 李阳阳, 焦李成. 量子协同免疫算法用于 SAT 问题的求解. 系统工程与电子技术, 2009, 31(6): 1441-1446.
- [75] Luiz Lenarth Vermaas, Leonardo M. Honorio, Muriel Freire, Daniele Barbosa. Learning Fuzzy Systems by a Co-Evolutionary Artificial-Immune-Based Algorithm. In Proceedings of the 8th International Workshop on Fuzzy Logic and Applications, Palermo, Italy, Springer, 2009: 312-219.
- [76] 武彦斌, 彭苏萍. 基于混沌免疫算法和遥感影像的土地利用分类. 农业工程学报, 2007, 23(6): 154-160.
- [77] 张海英, 韩贵金, 潘永湘. 混沌免疫进化算法及其在函数优化中的应用. 模式识别与人工智能, 2007, 20(2): 225-232.
- [78] 何宏, 钱锋. 基于免疫网络调节机理的混沌超变异免疫算法. 系统仿真学报, 2008, 20(11): 2810-2817.
- [79] 杨海东, 鄂加强. 自适应尺度混沌免疫优化算法及其应用. 控制理论与应用, 2009, 26(10): 1069-1074.
- [80] 阳阳, 焦李成. 求解 SAT 问题的量子免疫克隆算法. 计算机学报, 2007, 30(2): 176-183.
- [81] 高洪元, 刁鸣, 赵忠凯. 基于免疫克隆量子算法的多用户检测器. 电

- 子与信息学报, 2008, 30(7): 1567-1571.
- [82] 陶新民, 陈万海, 郭黎利. 一种新的基于模糊聚类 and 免疫原理的入侵监测模型. 电子学报, 2006, 34(7): 1329-1331.
- [83] 何宏, 钱锋. 基于模糊自适应免疫算法的非线性系统模型参数估计. 控制理论与应用, 2009, 26(5): 481-486.
- [84] 王辉, 王科俊, 于立君等. 基于模糊思想生成最有效检测器集的变阈值免疫算法. 系统仿真学报, 2008, 20(17): 4596-4601.
- [85] 宋晓琳, 于德介, 殷智宏. 采用免疫算法优化设计汽车主动悬架的模糊控制器. 系统仿真学报, 2006, 18(9): 2634-263.
- [86] 史婷娜, 张典林, 夏长亮等. 基于遗传整定的永磁交流伺服系统模糊免疫 PID 控制器. 电工技术学报, 2008, 23(7): 45-50.
- [87] 丁永生, 任立红. 一种新颖的模糊自调整免疫反馈控制系统. 控制与决策, 2000, 15(4): 443-447.
- [88] 曹先彬, 刘克胜, 王煦法. 基于免疫进化规划的多层前馈网络设计. 软件学报, 1999, 10(11): 1180-1184.
- [89] 刘志远, 吕剑虹, 陈来九. 新型 RBF 神经网络及在热工过程建模中的应用. 中国电机工程学报, 2002, 22(9): 118-122.
- [90] 王磊, 焦李成, 刘芳等. 免疫进化子波网络及其学习算法. 电子学报, 2001, 29(12A): 1878-1885.
- [91] 左兴权, 李士勇. 采用免疫进化算法优化设计径向基函数模糊神经网络控制器. 控制理论与应用, 2004, 21(4): 521-525.
- [92] 侯家利, 朱梅阶, 彭宏. 模块化免疫神经网络的模型研究. 电子学报, 2005, 33(8): 1502-1505.
- [93] 侯胜利, 毕宏, 毕志蓉, 王威. 一种基于神经网络的免疫识别故障检测模型. 系统仿真学报, 2009, 21(7): 1887-1891.
- [94] 侯水平, 焦李成, 田小林. 基于免疫克隆聚类协同神经网络的图像识别. 电子与信息学报, 2008, 30(2): 263-266.
- [95] 李伟, 郝建红, 祁兵. 一种利用 CPRNG 实现的混沌同步加密通信方案. 物理学报, 2008, 57(3): 1398-1402.
- [96] 徐淑奖, 王继志. 一类改进的混沌迭代加密算法. 物理学报, 2008, 57(1): 37-41.

- [97] 杨吉云, 廖晓峰, 肖迪等. 对一种基于 logistic 映射的分组加密机制的分析和改进. 通信学报, 2008, 29(12): 86-91.
- [98] Lipo Wang, Wen Liu, Haixiang Shi. Noisy Chaotic Neural Networks With variable Thresholds for the Frequency Assignment Problem in Satellite Communications. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, 2008, 38(2): 209-216.
- [99] Jovic, B., Unsworth, C. P. Fast synchronisation of chaotic maps for secure chaotic communications. Electronics Letters, 2010, 46(1): 49-50.
- [100] Pisarchik, A. N., Ruiz-Oliveras, F. R.. Optical Chaotic Communication Using Generalized and Complete Synchronization. IEEE Journal of Quantum Electronics, 2010, 46(3): 279-284.
- [101] 陈菊芳, 程丽, 刘颖. 延迟变量反馈法控制离散混沌系统的电路实验. 物理学报, 2003, 52(1): 18-24.
- [102] 李亚, 张正明, 陶志杰. 一个超混沌六阶蔡氏电路及其硬件实现. 物理学报, 2009, 58(1): 6818-6822.
- [103] 宁新宝, 卞春华, 王俊等. 心脏电活动过程的非线性分析. 科学通报, 2006, 51(7): 765-771.
- [104] E. Ott, C. Grebogi, J. Yorke. Controlling chaos, Phys. Rev. Lett., 1990, 64(11): 1196-1199.
- [105] 王永生, 孙瑾, 王昌金等. 变参数混沌时间序列的神经网络预测研究. 物理学报, 2008, 57(10): 6121-6129.
- [106] Jianquan Lu, Jinde Cao, Daniel W. C. Ho. Adaptive Stabilization and Synchronization for Chaotic Lur'e Systems with Time-Varying Delay. IEEE Transactions on circuits and systems—i: regular papers, 2008, 55(5): 1347-1356.
- [107] Alexander L. Fradkov, Boris Andrievsky, Robin J. Evans. Adaptive Observer-Based Synchronization of Chaotic Systems with First-Order Coder in the Presence of Information Constraints. IEEE Transactions on circuits and systems—i: regular papers, 2008, 55(6): 1685-1694.
- [108] H. K. Lam, Wing-Kuen Ling, Herbert Ho-Ching Iu, et al. Synchronization of Chaotic Systems Using Time-Delayed Fuzzy State-Feedback Control.

- IEEE Transactions on circuits and systems-i: regular papers, 2008,55(3): 893-903.
- [109] 王兴元, 孟娟. 基于 Takagi-Sugeno 模糊模型的超混沌系统自适应投影同步及参数辨识. 物理学报, 2009, 58(6): 3780-3786.
- [110] 王东风, 韩璞. 基于粒子群优化的混沌系统比例!积分!微分控制. 物理学报, 2006: 56(4)1644-1649.
- [111] 刘朝华, 张英杰, 章兢等. 基于免疫双态粒子群的混沌系统自抗扰控制. 物理学报, 2011, 60(1): 019501: 1-9.
- [112] 刘朝华, 章兢, 张英杰, 李小花. 一类不确定离散混沌系统的自抗扰控制器与小脑神经网络并行优化控制. 物理学报, 2011, 60(3): 030701: 1-9.
- [113] 韩京清. 自抗扰控制技术, 北京: 国防工业出版社, 2008.
- [114] 刘朝华, 章兢, 张英杰等. 基于免疫协同粒子群进化算法的永磁同步电机多参数辨识模型方法[J]. 自动化学报, 2012, 38(10): 1698-1708.
- [115] M. A. Rahman, D. M. Vilathgamuwa, M. N. Uddin, et al. Nonlinear control of interior permanent magnet synchronous motor. IEEE Trans. Industry Applications, 2003,(39)2: 408-416.
- [116] Rashed M., Macconnell P.F.A., Stronach A.F., et al. Sensorless indirect-rotor-field-orientation speed control of a permanent-magnet synchronous motor with stator-resistance estimation. IEEE Transactions on Industrial Electronics, 2007, 54(3):1664-1675.
- [117] Rahman K. M., Hiti S.. Identification of machine parameters of a synchronous motor. IEEE Transactions on Industry Applications, 2005, 41(2):557-565.
- [118] Khatounian F., Moreau S., Monmasson E., et al. Simultaneous Identification of the Initial Rotor Position and Electrical Parameters of a PMSM for a Haptic Interface. Proceedings of 12th International Power Electronics and Motion Control Conference, Portoroz :IEEE press,2006:276-281.
- [119] Piippo A., Hinkkanen M., Luomi J.. Adaptation of Motor Parameters in Sensorless PMSM Drives. IEEE Transactions on Industry Applications, 2009, 45(1):203-212.

- [120] Morimoto S., Sanada M., Yakeda Y.. Mechanical sensorless drives of IPMSM with online parameter identification. *IEEE Transactions on Industry Applications*, 2006, 42(5):1241-1248.
- [121] Senjyu T., Kuwae Y., Urasaki N., Uezato K.. Accurate parameter measurement for high speed permanent magnet. *Proceedings of IEEE Power Electronics Specialists Conference*, Vancouver, BC: IEEE Press, 2001(2):772-777.
- [122] Reigosa D., Briz F., Garcia P., Degner M.W.. Magnet temperature estimation in surface PM machines using high frequency signal injection. *IEEE Transactions on Industry Applications*, 2010, 46(4):1468-1475.
- [123] Liu K., Zhu Z. Q., Zhang Q., et al. Influence of inverter nonlinearity on parameter estimation in permanent magnet synchronous machines. In *Proceedings of International Conference on Electrical Machines*, Rome, Italy: IPCSIT, 2010:1-5.
- [124] Underwood S., Husain I.. On-line parameter estimation and adaptive control of permanent magnet synchronous machines. *IEEE Transactions on Industrial Electronics*, 2010, 57(7):2435-2443.
- [125] Liu K., Zhang Q., Chen J.T., Zhu Z.Q., Zhang J., Shen A.W.. Online multiparameter estimation of nonsalient-pole PM synchronous machines with temperature variation tracking. *IEEE Transactions on Industrial Electronics*, 2011, 58(5): 1776 – 1788.
- [126] 刘侃, 章兢. 基于自适应线性元件神经网络的表面式永磁同步电机参数在线辨识. *中国电机工程学报*, 2010, 30(30): 68-73.
- [127] 龚固丰, 章兢, 何昭晖等. 混合编码免疫算法在船舶载重计量的应用. *控制理论与应用*, 2009, 26(3): 349 -352.
- [128] Potter M. A, D. E Jong K. A. A cooperative coevolutionary approach to function optimization. In *Proceedings of the 3rd Parallel Problem Solving from Nature*. Berlin: Springer-Verlag, 1994: 249 –257.
- [129] 宋丹, 傅明. 基于多种群的自适应免疫进化计算. *控制与决策*, 2005, 20(11): 1251-1256.
- [130] 曹先彬, 罗文坚, 王煦法. 基于生物种群竞争模型的协同进化. *软件学*

- 报, 2001, 12(4): 556 -562.
- [131] 刘朝华, 章兢, 张英杰等. 竞争合作型协同进化免疫算法及其在旅行商问题中的应用. 控制理论与应用, 2010, 27(10): 1322-1330.
- [132] Chi-keong Goh, Kay chen Tan. A competitive-cooperative coevolutionary paradigm for dynamic multi objective optimization. IEEE Transactions on Evolutionary Computation, 2009, 13(1): 103- 127.
- [133] 刘朝华, 张英杰, 李小花等. 双态免疫优势蚁群算法及其在 TSP 中的应用研究. 小型微型计算机系统, 2010, 31(5): 937-941.
- [134] Aras N, Altinel I. K, Oommen B.J.A. Kohonen-like decomposition method for the euclidean traveling salesman problem- KNIES DECOMPOSE. IEEE Transactions on Neural Networks, 2003, 14(1): 869 -890.
- [135] Vieira F. C, Neto A. D. D. An efficient approach to the traveling salesman problem using self-organizing maps. International Journal of Neural Systems, 2003, 13(2): 59 -66.
- [136] 张军英, 周斌. 基于泛化竞争和局部渗透机制的自组织网 TSP 问题求解方法. 计算机学报, 2008, 31(2): 220- 227.
- [137] 朱文兴, 傅清祥. 一个基于填充函数变换的对称 TSP 问题的局部搜索算法. 计算机学报, 2002, 25(7): 701-707.
- [138] Alice R. Malisia, Hamid R. Tizhoosh applying oppositionbased ideas to the ant colony system. In Proceedings of IEEE Swarm Intelligence Symposium (SIS2007). Hawaii, USA: IEEE Press, 2007: 182 -189.
- [139] Eberhart R, Kennedy J. A. A new optimizer using particle swarmtheory. In Proceeding of International Symposium on Micromachine and Human Science. Nagoya, Japan: IEEE Press, 1995: 39 -43.
- [140] 刘朝华, 张英杰, 章兢等. 一种双态免疫粒子群算法.控制理论与应用, 2011, 28(1): 65-72.
- [141] C. F. Juang.A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Trans. Syst., Man, Cybern. B,Cybern, 2004,34(2): 997-1006.
- [142] A. A. E. Ahmed, L. T. Germano, Z. C. Antonio.A hybrid particle swarm optimization applied to loss power minimization. IEEE Trans.Power Syst,

- 2005, 20(2): 859-866.
- [143] S.H. Ling, H.H.C. Ju, K.Y. Chan, et al. Hybrid Particle Swarm Optimization With Wavelet Mutation and Its Industrial Applications. *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, 2008, 38(3): 743-763.
- [144] Y. Shi, R. C. Eberhart. A modified particle swarm optimizer, In *Proc. IEEE World Congr. Comput. Intell.*, Anchorage, AK: IEEE Press, 1998: 69-73.
- [145] Y. Shi, R. C. Eberhart. Particle swarm optimization with fuzzy adaptive inertia weight, In *Proc. Workshop Particle Swarm Optimization*, Indianapolis, IN: IEEE Press, 2001: 101-106.
- [146] Z.H. Zhan, J. Zhang, Y. Li, et al. Adaptive Particle Swarm Optimization, *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, 2009, 39(6): 1362-1381.
- [147] J. J. Liang, A. K. Qin, P. N. Suganthan, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.*, 2006, 10(3): 281-295.
- [148] H. Wu, J. P. Geng, R. H. Jin, J. Z. Qiu, et al. An Improved Comprehensive Learning Particle Swarm Optimization and Its Application to the Semiautomatic Design of Antennas. *IEEE Trans. Antennas and Propagation.*, 2009, 57(10): 3018-3028.
- [149] 王瑜, 李斌, 袁博. 混合 SQP 的基于完全学习的粒子群优化算法在电力系统中经济分配问题的应用. *中国科学: 信息科学*, 2010, 40(3): 403-411.
- [150] J. Kennedy, R. Mendes. Population structure and particle swarm performance. In *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI: IEEE Press, 2002: 1671-1676.
- [151] J. Kennedy, R. Mendes. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, 2006, 36(4): 515-519.
- [152] Ho S. L., Shi Y. Y., Guang Z. N., et al. A particle swarm optimization based method for multiobjective design optimizations. *IEEE Transactions on Magnetics*, 2005, 41(5): 1756-1759.
- [153] 任子武, 伞冶. 实数遗传算法的改进及性能研究. *电子学报*, 2007, 35(2): 270-271.

- [154] 李宏, 唐焕文, 郭崇慧. 一类进化策略的收敛性分析. 运筹学学报, 1999, 3(4): 79-83.
- [155] 郭崇慧, 唐焕文. 演化策略的全局收敛性. 计算数学, 2001, 23(1): 105 -110.
- [156] A. A. E. Ahmed, L. T. Germano, and Z. C. Antonio. A hybrid particleswarm optimization applied to loss power minimization. IEEE Trans. Power Syst., 2005, 20(2) :859-866.
- [157] 张梅凤, 邵诚. 多峰函数优化的生境人工鱼群算法. 控制理论与应用, 2008, 25(4): 773 -776.
- [158] Zhao-Hua Liu, Jing Zhang, Shao-Wu Zhou, et al. Co evolutionary Particle Swarm Optimization Using AIS and Its Application in Multi-parameter estimation of PMSM. IEEE Transactions on Cybernetics, 2013, 43(6): 1921-1935.
- [159] F. Van den Bergh a, A. P. Engelbrecht. A cooperative approach to particle swarm optimization. IEEE Trans. Evol. Comput., 2004, 8(3): 225-239.
- [160] G.G. Yen, W.F. Leong. Dynamic Multiple Swarms in Multiobjective ParticleSwarm Optimization. IEEE Trans. Syst. Man Cybern. Part A-Syst. Hum., 2009, 39(4): 890-911.
- [161] B. Niu, Y. Zhu, X. He. Multi-population cooperative particle swarm optimization, in Proc. Eur. Conf. Artif. Life, Canterbury, U.K., IEEE Press, 2005: 874-883.
- [162] 高芳, 崔刚, 吴智博. 求解复杂优化问题的双层多种群粒子群优化算法. 高技术通讯, 2009, 19(5): 519-524.
- [163] 李丹, 高立群, 王珂. 基于动态多种群粒子群支持向量机的短期负荷预测. 计算机科学, 2008, 35(7): 133-136.
- [164] 赵波, 郭创新, 张鹏翔等. 基于分布式协同粒子群优化算法的电力系统无功优化, 中国电机工程学报, 2005, 25(21): 1-7.
- [165] 张勇, 巩敦卫, 张婉秋. 一种基于单纯形法的改进粒子群优化算法及其收敛性分析. 自动化学报, 2009, 35(3): 289-298.
- [166] J. J. Liang, P. N. Suganthan, K. Deb. Novel composition test functions for numerical global optimization. In Proc. IEEE Swarm Intell. Symp.,

- Pasadena, CA:IEEE Press,2005:68-75.
- [167] Marco Dorigo , Gianni Di Caro. Ant Algorithms for Discrete Optimization. *Artificial Life*, 1999, 5 (3): 137- 172.
- [168] 刘朝华, 张英杰, 章兢等. 蚁群算法与免疫算法的融合及其在 TSP 中的应用. *控制与决策*, 2010, 25(5): 695-700, 705.
- [169] 戚玉涛, 刘芳, 焦李成. 求解 TSP 问题免疫算法的动态疫苗策略. *西安电子科技大学学报*, 2008, 5(1): 37-42.
- [170] 王磊, 潘进, 焦李成. 免疫算法. *电子学报*, 2000, 28 (7): 74-78.
- [171] 段海滨. 蚁群算法原理及其应用. 北京: 科学出版社, 2006, 45-96.
- [172] Alice R. Malisia, Hamid R. Tizhoosh Applying Opposition-Based Ideas to the Ant Colony System In: proceedings of IEEE Swarm Intelligence Symposium(SIS2007), Hawaii,USA:IEEE Press, 2007:182-189.
- [173] 甄子洋, 王志胜, 王道波. 基于信息融合最优估计的非线性离散系统预测控制. *自动化学报*, 2008, 34(3): 331-336.
- [174] Peter Shih, Brian C. Kaul, Sarangapani Jagannathan,etal. Reinforcement-Learning-Based output-FeedbackControl of Nonstrict Nonlinear Discrete-TimeSystems With Application to Engine Emission Control.IEEE Transactions on Systems, Man, and Cybernetics—Part B: ybernetics, 2009, 39(5): 1162-1178.
- [175] 任海鹏, 刘丁. 六辊 UC 轧机轧制过程混沌的轨迹跟踪控制. *控制理论与应用*, 2002, 19(6): 897-900.
- [176] 张文革, 韩京清. 一类混沌系统的状态观测与控制. *控制与决策*, 2000, 15(3) 301-304.
- [177] 邵立伟, 廖晓钟, 张宇河. 基于时间尺度的感应电机自抗扰控制器的参数整定. *控制理论与应用*, 2008: 25(2): 205-209.
- [178] 史永丽, 侯朝桢, 苏海滨. 基于粒子群优化算法的自抗扰控制器设计. *系统仿真学报*, 2008, 20(2): 433-436.
- [179] 辛菁, 刘丁, 杨延西等. 基于自适应免疫整定的机器人无标定自抗扰视觉伺服控制. *控制理论与应用*, 2007: 24(4): 546-552.
- [180] Albus J. S. A New Approach to Manipulator Control,the Cerebellar Model Articulation Controller (CMAC). *Transactions of the ASME Journal of*

- Dynamic System Measurement and Control, 1975, 97(3): 220-227.
- [181] Chih-Min Lin, Te-Yu Chen. Self-Organizing CMAC Control for a Class of MIMO Uncertain Nonlinear Systems. IEEE Transactions on neural networks, 2009, 20(9): 1377-1383.
- [182] 刘金琨. 先进 PID 控制 MATLAB 仿真. 北京: 电子工业出版社, 2007.
- [183] 温淑焕. Hénon 混沌系统的自适应预测函数控制快速算法. 物理学报, 2009, 58(8): 5209-5213.
- [184] 侯忠生, 王卫红, 金尚泰. 一类非线性离散系统自适应准滑模控制. 控制理论与应用, 2009, 26(5): 505-509.
- [185] Ramakrishnan R, Islam R, Islam M, et al. Real time estimation of parameters for controlling and monitoring permanent magnet synchronous motors . In Proceedings of IEEE International Electric Machines and Drives Conference, Miami, USA: IEEE Press, 2009: 1194 -1199.
- [186] S. Bolognani, L. Tubiana, M. Zigliotto. Extended Kalman filter tuning in sensorless PMSM drives. IEEE Trans. on Industry Applications, 2003, 39(6): 1741-1747.
- [187] T. Liu, M. Elbuluk, I. Husain. Sensorless adaptive neural network control of permanent magnet synchronous motors. In Proceedings of International Conference on Electric Machines and Drives, Seattle, WA, USA : IEEE Press, 1999: 287-289 .
- [188] Rashed M, Macconnell P. F. A, Stronach A. F, et al. Sensorless indirect-rotor-field- orientation speed control of a permanent-magnet synchronous motor with stator-resistance estimation. IEEE Transaction on Industrial Application, 2007, 54(3): 1664-1675.
- [189] Kan Liu, Qiao Zhang, Zi-Qiang Zhu, et al. Comparison of Two Novel MRAS Based Strategies for Identifying Parameters in Permanent Magnet Synchronous Motors. International Journal of Automation and Computing, 2010, 7(4): 516-524.
- [190] K.M. Rahman, S. Hiti. Identification of machine parameters of a synchronous motor. IEEE Trans. on Industry Applications, 2005, 41(2): 557-565.

- [191] Zhao-Hua Liu, Jing Zhang, Xiaohua Li, etal. Cooperative Particle Swarm Optimization with ICS and Its Application to Parameter identification of PMSM. The 7th IEEE Conference on Industrial Electronics and Applications (ICIEA 2012), Singapore, 2012, 1303-1308.
- [192] Li Liu, Wenxin Liu, David A. Cartes. Permanent Magnet Synchronous Motor Parameter Identification using Particle Swarm Optimization. International Journal of Computational Intelligence Research. 2008, 4(2): 211-218.
- [193] 许伯强, 李和明, 朱凌. 发电机定转子绕组温度在线监测新方法. 电力系统自动化, 2002, 35(10):35-39.
- [194] 王立国, 张凤娜, 吕辛. 基于热力学参数摄动分析的潜油电机温度辨识. 电工技术学报, 2011, 26(6): 7-11.
- [195] 郭科, 陈聆, 魏友华. 最优化方法及其应用. 北京: 高等教育出版社, 2007.